"صخرة الأساس"

BASIC ROCK

ملخص لغة Ring

تأليف: احمد حسونة عام 2019

نبذه مختصرة

هذا الكتاب يعد للقارئ بمثابة صخرة قوية ثابته بفضل الله من الأساس في عالم لغة البرمجة رينج، مع ملخص سريع في لغة الـ Ring السهلة والجميلة والقوية مع أهم تفاصيلها من البداية بفضل الله، حيث أن هناك كود وتنفيذ لتطبق، ثم تتعلم أكثر، ثم تطبق أكثر، ثم تتأسس وتتعلم أكثر وأكثر، ثم تمارس أكثر وتستمر في التطبيق، ثم تصل الى الاحتراف بفضل الله

احمد حسونة

صخرة الأساس ملخص Ring

فهرس الكتاب

المحتويات

فهرس الكتاب 1
مقدمة أساسية – هامة جداً 6
هام جداً Very Important قبل الكتاب 10
شرح مرئي للكتاب 11
قواعد 11
مميزات وامكانيات لغة البرمجة Ring 12
تنزيل اللغة وتشغيلها – ويندوز، لينكس، ماك 13
الطباعة – الدالة see 14
تنفيذ أكثر من أمر بسطر واحد 15
دمج النصوص بالرمز + ... 15
المعاملات الحسابية 16
المتغيرات 16
طباعة نوع البيانات – الدالة type 18
متغير نصي متعدد الأسطر 19
المتغير القائمة List 20
متغير القوائم المتداخلة 25
المتغير القاموس او الـ Hash Table 27
عدد الأشياء – الدالة len 28
التعليقات التي لا تنفذ مع الكود 29

عوامل التخصيص 30
التحويل من رقم لنص – الدالة string 31
التحويل من نص لرقم – الدالة number 32
تحويل الحرف الى ASCII – الدالة ascii 32
تحويل الـ ASCII لحرف – الدالة char 33
استلام من المستخدم – الدالة give 33
العشوائية – الدالة random 35
العوامل المنطقية بالكلمات and, or, not 37
عوامل المقارنة 38
تحويلات النظم العددية 39
تحميل المكتبات - load 40
تقطيع النص – الدالة substr 40
استبدال النص – الدالة substr 42
تحولات النص الكبير والصغير ABC, abc 42
تحقق النص 43
اتخاذ القرار – جملة if 44
حلقات التكرار – الجملة for 48
التكرار المتداخل nested loop 54
حلقات التكرار – الجملة while 57
حلقات التكرار – الجملة do... again 59
التكرار اللانهائي 60
الخروج من التكرار - exit 61

الاستكمال في التكرارات - loop 62
الوقت والتاريخ – date, time, timelist 63
تخصيص التاريخ والوقت 65
التلاعب مع التاريخ والوقت 65
برنامج حساب العمر 66
فتح ملف نصي موجود لتجهيزه للقراءة 67
انشاء ملف نصي فارغ وتجهيزه للكتابة 67
انشاء ملف نصي والكتابة عليه 68
القراءة من ملف موجود 69
قراءة الملف بالكامل بالتكرار بأكثر من طريقة 71
الكتابة على الملف بسهولة – الدالة write 73
القراءة من الملف بسهولة – الدالة read 73
القراءة من ملف من مكان معين 74
القراءة من ملف سطر تلو السطر 74
تكملة الكتابة على ملف نصي موجود 75
تكملة الكتابة مع القراءة من ملف نصي 76
القراءة مع الكتابة لملف نصي 77
الكتابة مع القراءة 77
القراءة والكتابة والتكملة لملف binary 78
التحقق من وجود ملف او مجلد وانشاء المجلدات 79
حذف الملفات والمجلدات 79
معرفة الملفات والمجلدات الموجودة بمجلد معين 80

تنفيذ أوامر الدوس 81
نسخ الملفات - copy 81
نسخ المجلدات 81
معالجة الأخطاء 82
ترتيب الكود في لغة البرمجة Ring – هام جداً 83
انشاء دوال functions عادية لا تنفذ شيء 84
انشاء دوال بها اكواد ثم النداء عليها في الأعلى 84
انشاء دالة تستقبل وسائط 85
انشاء دالة ترجع قيمة 88
ارسال الدالة كوسيط ثم تشغيلها 89
انشاء دالة تشغل الدوال بعدد من المرات 90
استخدام الدوال المجهولة أو تعبيرات Lambda 91
انشاء دالة مرجعية تنادي على نفسها 92
انشاء مكتبة أكواد قابلة للتنفيذ 93
تنفيذ أكواد من ملف خارجي أو من نص 96
انشاء class به خصائص فقط 98
انشاء class به وظائف فقط 99
انشاء class به خصائص ووظائف معاً 99
انشاء class واستخدامه 100
انشاء class وعمل أكثر من object له 101
استخدام Regular Expression 102
استخدام الرياضيات math 103

دالة البناء في الكلاس 105
وسائط مع دالة البناء 106
انشاء عداد وindex للـ objects 107
انشاء عناصر سرية داخل الـ class 108
الوراثة 109
اظهار اسم الكلاس 110
إعادة الكتابة على الدوال 110
الحزم Packages 112
موضوع Ring Package Manager مفيد جداً 113
وبعض البرامج التي تم عملها بـ Ring 113
وبعض الألعاب Games التي تم عملها بـ Ring 113
كنوز وهدايا 💖 116
شكر خاص 127
نشكرك على قراءتك 128
المراجع References 129
الختام The End 130

مقدمة أساسية – هامة جداً

بسم الله الرحمن الرحيم والصلاة والسلام على أشرف المرسلين سيدنا محمد الذي أرسله الله رحمة للعالمين، وأشهد أن لا إله إلا الله وحده لا شريك له، الذي لم يتخذ ولدا ولم يكن له شريك في الملك وخلق كل شيء فقدره تقديرا ،،، ثم أما بعد...

الشيء الوحيد الذي أخبر الله رسوله محمد ﷺ أن يطلب فيه الزيادة هو العلم عندما قال الله سبحانه وتعالي في سورة طه (وَقُلْ رَبِّ زِدْنِي عِلْمًا) ، وأول ما نزل من القرآن الكريم (اقْرَأْ بِاسْمِ رَبِّكَ الَّذِي خَلَقَ (1) خَلَقَ الْإِنْسَانَ مِنْ عَلَقٍ (2) اقْرَأْ وَرَبُّكَ الْأَكْرَمُ (3) الَّذِي عَلَّمَ بِالْقَلَمِ (4) عَلَّمَ الْإِنْسَانَ مَا لَمْ يَعْلَمْ) ، ورفع الله من قدر وشأن العلماء حينما قال (يَرْفَعِ اللَّهُ الَّذِينَ آمَنُوا مِنكُمْ وَالَّذِينَ أُوتُوا الْعِلْمَ دَرَجَاتٍ) ، وبين الله عز وجل أن العلماء لا يتساوون في القدر مع من لا يعلم فقال جل في علاه (قُلْ هَلْ يَسْتَوِي الَّذِينَ يَعْلَمُونَ وَالَّذِينَ لَا يَعْلَمُونَ إِنَّمَا يَتَذَكَّرُ أُولُو الْأَلْبَابِ) ، وأيضاً أخبرنا رب العزة سبحانه أن من يخشونه هم العلماء حين قال في كتابه العزيز (إِنَّمَا يَخْشَى اللَّهَ مِنْ عِبَادِهِ الْعُلَمَاءُ) ، وهو الذي علمنا عدد السنين والحساب حين قال (هُوَ الَّذِي جَعَلَ الشَّمْسَ ضِيَاءً وَالْقَمَرَ نُورًا وَقَدَّرَهُ مَنَازِلَ لِتَعْلَمُوا عَدَدَ السِّنِينَ وَالْحِسَابَ مَا خَلَقَ اللَّهُ ذَلِكَ إِلَّا بِالْحَقِّ يُفَصِّلُ الْآيَاتِ لِقَوْمٍ يَعْلَمُونَ) ، وكما نري أن العلم قدرة عالي ولا يطلبه إلا عالي أدرك علاه.

ولقد عزمت بإذن الله تعالي أن أقدم أفضل ما لدي من علوم في مجالات الحاسب الآلي (الكومبيوتر)، لأنه يفيد في جميع مناحي الحياة، فإذا تحدثنا عن شركة تجارية فهي تستخدم الكمبيوتر لتدوين وحساب جميع اعمالها من انتاج وتسويق ورواتب وغيرها، وإذا تحدثنا عن مدرسة أو كلية أو معهد، فجميعها تستخدم الكمبيوتر في أعمالها وتظهر نتائج الامتحانات والجداول، بل وبعض المؤسسات التعليمية الحديثة تمتحن الطالب ببرامج ذكية متطورة، وفي الهيئات الحكومية يتم استخدام الكمبيوتر أيضاً، وأصبح الحاسب يفعل كل شيء في حياتنا فلقد صعد الإنسان في الآفاق والفضاء بالحاسب، ونشاهد به الفيديو ونشغل به الصوت ونطبع به الوثائق والمستندات ونصور به الصور والأوراق وندير به كافة المجالات، ولا نقتصر على هذا فحسب، بل ولو قلت لك أن الحاسب قد يغير في حياة أسرة بأكملها وينجيها بفضل الله ، سوف تتعجب وتقول كيف ؟! – والجواب هو كم من رجل كان على حافة الانهيار المادي والمعنوي بسبب فقده لوظيفته، ففتح الله علية بتعلم الحاسب ورزقه الله بعمل جيد ورزق وفير، وكم من شاب كان على حافة الانحراف الأخلاقي بسبب الأخلاق السيئة المتداولة بين الشباب إلا من رحم ربي، فمن الله علية بأن بدء في تعلم مجالات الحاسب فأحبها وبدء يبني مستقبلة الحقيقي.

المحتوي الذي سوف نتعلمه يتكون من مواضيع تتبع بعضها بعض، ومنظمة بطريقة تسلسلية بحيث تأخذ بيدك خطوة بخطوة من البداية إلى الاحتراف بفضل الله، وسوف أتبع نظام في شرح هذا المحتوي مثل صعود الطائرة، ففي البداية يتم التنبيه على الركاب ببعض الإرشادات بالتفصيل الدقيق، ثم

في بداية الإقلاع يكون الأمر صعباً ثم نطير إلى أعلي ونحلق في الفضاء، وسوف نبدأ بإذن الله بالتفصيل ثم نعلوا.. ونعلوا ، ثم نحلق ونحترف بإذن الله، فإنني أنصح القارئ أن يهتم بجميع الموضوعات مرتبة، فعسي كلمة أن تغير بفضل الله، والمحتوي تم تأليفه بكامل الحب والإخلاص لجميع القراء، ولن يأخذ نسخ ولصق، بل وحتي المحتويات التاريخية الموجودة داخل المحتوي تم صياغتها بأساليب مختلفة من أكثر من مكان معلوماتي، والمؤلف وضع كل خبرته بكل الحب لجميع أحبائه القارئين لهذا المحتوي، والكمال لله وحدة، ونحن لا نعلم إلا قليلاً كما قال الله في القرآن (وَيَسْأَلُونَكَ عَنِ الرُّوحِ ۖ قُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا).

وفي الختام أسأل الله تبارك وتعالي بأسمائه الحسني وصفاته العلا أن يفيد جميع القراء بهذا المحتوي، ويغير في حياتهم إلى الأفضل إن شاء الله.

كما أحب أن اشجع الجميع على العطاء في العلم، فإذا كنت تعلم أي معلومة لا تبخل على غيرك بها أبداً، وتأكد أن الله سوف يكرمك بذلك ويمن عليك بمعلومات أفضل بكثير من الذي لديك، ولقد أمرنا رسول الله كامل الأخلاق أن نحب الخير لإخواننا كما نحبه لأنفسنا حين قال (حب لأخيك ما تحب لنفسك)، وهذا الموضوع هام للغاية لأنه ينهض بالمجتمع ككل، فإذا تعلمت شيء وعلمته لأخيك أو زميلك فسوف يأخذ فكرتك ثم يبتكر عليها ابتكارا مختلفاً من ابتكارك، لأن الله خلق البشر بعقول ومميزات مختلفة، ثم يعلم زميلك معلومتك ومعلومته لزميل آخر، فيبتكر ابتكاراً ثالثاً مختلفاً عن الأول والثاني، وهكذا، ولو حدث هذا على مستوي المجتمع

نهض المجتمع وكثرت ابتكاراته واختراعاته، ولا أقصد مجال الحاسب الآلي فحسب، لا، بل أقصد جميع المجالات في جميع مناحي الحياة، وإذا قابلت شخص طلبت منه معلومة وبخل عليك، ثم بعد ذلك طلب منك معلومة لا تفعل مثلة، بل اعطية المعلومة على الفور ولمح له أن هذا هو الصحيح، وسوف يؤثر هذا علية ويغير حالة إلى العطاء، وهذا حدث معي شخصياً أكثر من مرة، وأثبتت لي التجربة أن هذا هو الصحيح.

كما أحب أن انوه أنه غير مسموح لأي فرد أو شركة أو هيئة أو منظمة أو مؤسسة أو قطاع من أي نوع أن يتربح أو يتاجر بهذا المحتوي بأي نوع أي كان، أو يعيد كتابته، أو ينسخه بغير حق ليغير الاسم، أو أي عملية تحويل كتاب pdf الى أي نوع أو التعديل عليه، ولن اسامح أي شخص تسبب في ذلك، ولن أتركه يوم القيامة، وعند الله تجتمع الخصوم.

وسوف تجد شرح فيديو لهذا المحتوي بالتفصيل مجاناً للمشاهدة **فقط** على أكاديمية حسونة في الموقع أو اليوتيوب على الرابطين التاليين:

www.hassouna-academy.com

www.youtube.com/user/hassounaacademy

والسلام عليكم ورحمة الله وبركاته.

المؤلف: احمد حسونة

هام جداً Very Important قبل الكتاب

أمانة - ثمن الكتاب، قراءة هذه الصفحة جيداً وصفحة نشكرك على قراءتك وصفحة مقدمة أساسية وتلبيه ما فيهم 😊 فضلاً منكم وليس أمراً.

موقع حسونة أكاديمي ← https://www.hassouna-academy.com

قـناة حسونة أكاديمي ← https://www.youtube.com/user/HassounaAcademy

بفضل الله، القناة عليها شرح كمبيوتر، ووينـدوز، ولينكس، وماك، وورد، واكسل، وأكسس، وباور بوينت، ولغات برمجة كثيرة، وويب بتقنيات كثيرة، وأمثلة كثيرة، وتطبيقات كثيرة، وموبايل، وأندرويد، وقواعد بيانات، ونظم، وتصميم، وتحليل بيانات، وتحليل نظم، ومناهج دراسية، ووسائط متعددة، ودردشة برمجة، وبرنامج مبيعات، وبرنامج مطاعم، ومعلومات متعددة، وقصص نجاح، وبرامج جاهزة، وفوتوشوب، وشروحات كثيرة بفضل الله.

رجاء! لا تنسي أن تشترك بالقناة ↗ وفعل زر الجرس في القناة 🔔 وسجل حساب على موقع حسونة أكاديمي 👤 الآن.

https://www.hassouna-academy.com/

شرح مرئي للكتاب

يوجد شرح مرئي على أكاديمية حسونة لهذا الكتاب من الرابط التالي:
https://youtu.be/rOwOEFTVvg0
وهو مقطع فيديو حماسي لتعليم الرينج. ونصيحة لك لا يفوتك أبداً!

قواعد

- اكتب بجوجل Ring Programming Language للمعرفة
- لغة رينج لا تفرق بين الحروف الكبيرة والصغيرة مثل A وa
- لغة رينج تجعل نوع المتغير بناء على قيمته التي اعطيتها إياه
- من قوة لغة Ring أن لها أنماط Styles كثيرة في كتابة الأكواد، حيث أنك قد تكتب الكود بطريقة معينة، ثم تجد أن نفس الكود يتم كتابته بطريقة مختلفة ويؤدي نفس الناتج، وهذه مرونة شديدة
- لابد من التركيز جيداً في الكود وفي تنفيذ الكود لأن بهم بعض التفسير
- اعلم أن هذا الكتاب بمثابة مراجعة سريعة على لغة الرينج بدون تعليقات كثيرة على الأكواد، حيث أن كود لغة الرينج من أسهل الأكواد التي يمكنك كتابتها وفهم الكود فقط من قراءة عنوان الدرس في الكتاب
- إذا كنت مبتدئ يفضل ان تشاهد شرح الرينج الموجود على قناة حسونة أكاديمي أولاً، لتعرف اللغة وتتعلم جيداً قبل مراجعتها من هذا الكتاب

مميزات وامكانيات لغة البرمجة Ring

- يوجد العديد من لغات البرمجة الديناميكية الناجحة التي تتميز بالسهولة مثل لوا Lua التي تتميز بصغر الحجم، ومثل بايثون Python التي تتميز بالبساطة والقوة، ومثل روبي Ruby التي تتميز بالمرونة، ولغة الرينج Ring تجمع لك كل هذه المزايا مع بعضها البعض، حيث أن لغة البرمجة Ring سهلة وبسيطة وقوية ومرنة وصغيرة الحجم ومتعددة الأغراض، وأكثر من ذلك الكثير والكثير، حيث أنها تقدم وتضيف أشياء جديدة كثيرة جداً
- تنزيل لغة البرمجة رينج من الانترنت سهل جدا ← ادخل على جوجل وابحث عن Ring Programming Language
- تشغيل لغة الرينج على نظامك سواء windows او Linux او Mac سهل جدا وتم شرحه على القناة من قبل
- لغة رينج تقدم لك برنامج جميل تكتب فيه أكواد الرينج وتشغلها وتصمم منه الفورم كما تحب، وهذا البرنامج تم عمله بلغة رينج نفسها وأكواد هذا البرنامج مفتوحة المصدر وتستطيع تفتح ملفاتها وتتعلم منها في اي وقت
- لغة البرمجة Ring تمكنك من عمل تطبيق ويب وموبايل وسطح مكتب
- لغة البرمجة Ring تمكنك من ربط برنامج بقواعد البيانات
- لغة البرمجة Ring تمكنك من عمل برنامج يعمل على نظام ويندوز ونفس البرنامج يعمل على نظام لينكس ونفس البرنامج يعمل على نظام الماك ونفس البرنامج يعمل على الموبايل أندرويد وايفون

- لغة البرمجة Ring تمكنك من عمل العاب 2D و3D بسهولة
- لغة Ring تعمل كـ declarative programming
- لغة Ring تعمل كـ natural programming
- لغة Ring تعمل كـ imperative programming
- لغة Ring تعمل كـ procedural oriented programming
- لغة Ring تعمل كـ object oriented programming
- لغة البرمجة Ring تمكنك من عمل كل ما تريد واكثر بفضل الله
- لغة البرمجة Ring تسمح بالربط مع مكتبات وادوات C و C++
- لغة البرمجة Ring لها بناء مستقل خاص بها
- لغة البرمجة Ring لها مترجم Compiler مستقل خاص بها
- Ring لها آلة Virtual Machine مستقلة خاصة بها
- لغة Ring تحتوي على أدوات تسهل على المبرمجين والمطورين مثل RingPM وRing2EXE وغيرها الكثير والكثير بفضل الله

تنزيل اللغة وتشغيلها – ويندوز، لينكس، ماك

يمكنك الرجوع الى الرابط التالي لتنزيل اللغة وتشغيلها على نظام ويندوز
https://youtu.be/ThPcTPyTkXc
ويمكنك الرجوع الى الرابط التالي لتنزيل اللغة وتشغيلها على نظام لينكس
https://youtu.be/nvgzIfuOnFo
ويمكنك الرجوع الى الرابط التالي لتنزيل اللغة وتشغيلها على نظام الماك
https://youtu.be/ibdTtHm_61I

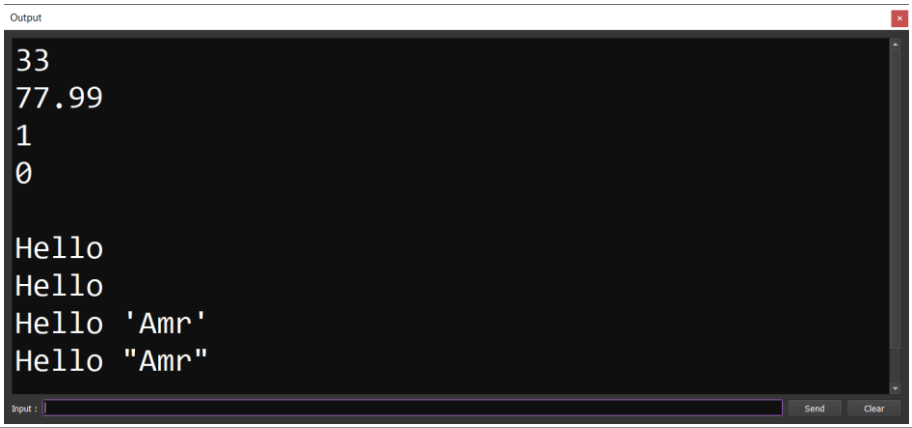
الطباعة – الدالة see

لاحظ! أنه فيما يلي تم طباعة قيم مختلفة بالكلمة see ولاحظ انه تم طباعة سطر جديد بـ NL

الكود

```
see 33
see NL
see 77.99
see NL
see True
see NL
see False
see NL
see NULL
see NL
see "Hello"
see NL
see 'Hello'
see NL
see "Hello 'Amr'"
see NL
see 'Hello "Amr"'
```

التنفيذ



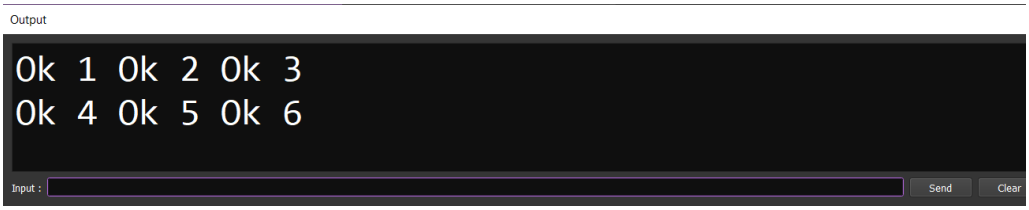
تنفيذ أكثر من أمر بسطر واحد

لاحظ! أنه فيما يلي تم تنفيذ أكثر من جملة برمجية باستخدام مسافة للعزل بين كل جملة وأخري

الكود

```
see 'Ok 1 ' see 'Ok 2 ' see 'Ok 3' see nl
see( NL + 'Ok 4 ') see( NL + 'Ok 5 ') see( NL + 'Ok 6')
```

التنفيذ



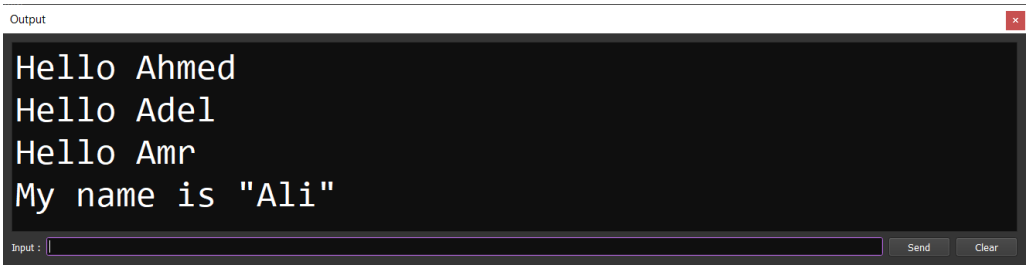
دمج النصوص بالرمز +

لاحظ! أنه فيما يلي تم دمج النصوص ولصقها بالرمز + بين النصوص

الكود

```
see "Hello" + " " + "Ahmed" + nl
see "Hello" + ' ' + 'Adel' + nl
see 'Hello' + ' ' + 'Amr' + nl
see 'My' + " " + 'name is ' + '"' + "Ali" + '"'
```

التنفيذ



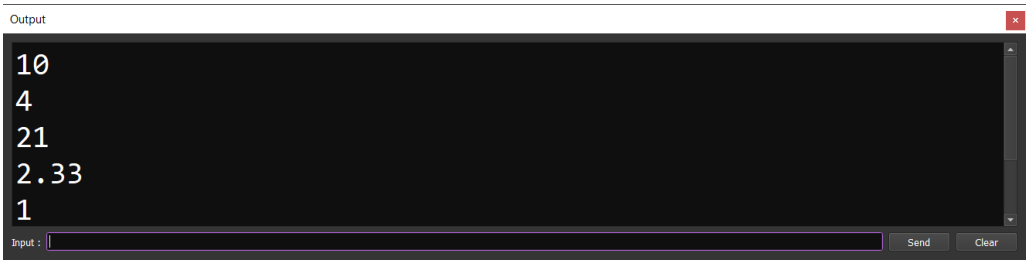
المعاملات الحسابية

لاحظ! أنه فيما يلي تم استخدام الجمع والطرح والضرب والقسمة وباقي القسمة

الكود

```
see 7+3 see NL
see 7-3 see NL
see 7*3 see NL
see 7/3 see NL
see 7%3 see NL
```

التنفيذ



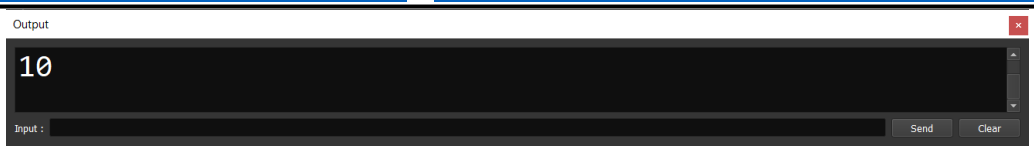
المتغيرات

لاحظ! أنه فيما يلي تم تعريف متغيرين ثم جمعهم في متغير ثالث

الكود

```
num1 = 7
num2 = 3
result = num1 + num2
see( NL + result)
```

التنفيذ

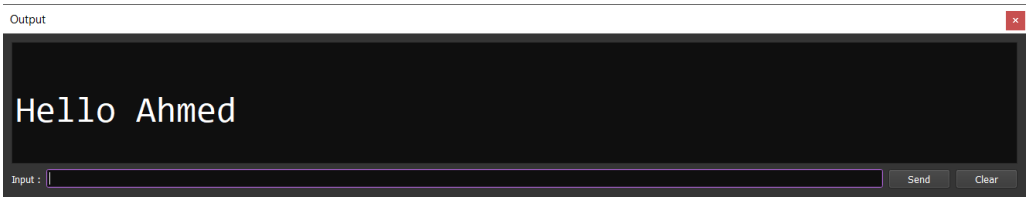


لاحظ! أنه تم عمل متغير ثم دمجه مع الكلمة Hello مسافة

الكود

```
name = 'Ahmed'
say_hello = 'Hello ' + name
see( NL + say_hello)
```

التنفيذ



لاحظ! أنه فيما يلي تم تعريف متغيرات واعطاؤهم القيم مع التعريف

الكود

```
name1='Ahmed' name2='Adel' name3 ='Amr'
see( NL +  name1 + ' ' + name2 + ' ' + name3 )
```

التنفيذ

الكود

```
num=1 name='Ali' salary=4600 is_active=True
```

```
see( NL +  num )
see( NL +  name )
see( NL +  salary )
see( NL +  is_active )
```

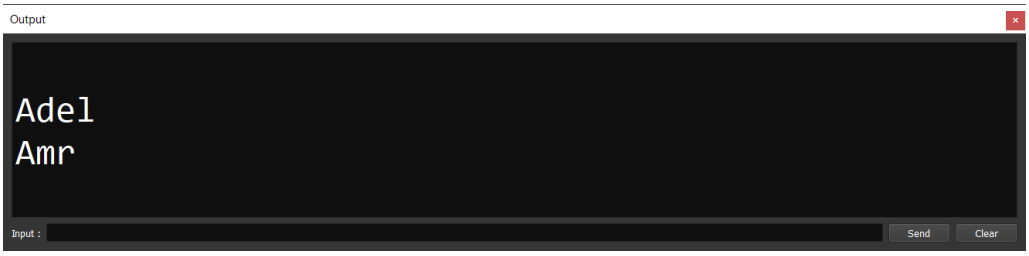
التنفيذ

لاحظ! أنه فيما يلي المتغير لا يحمل بداخلة الا قيمة واحدة فقط

الكود

```
name = 'Adel'
see( NL + name)
name = 'Ahmed'
name = 'Amr'
see( NL + name)
```

التنفيذ



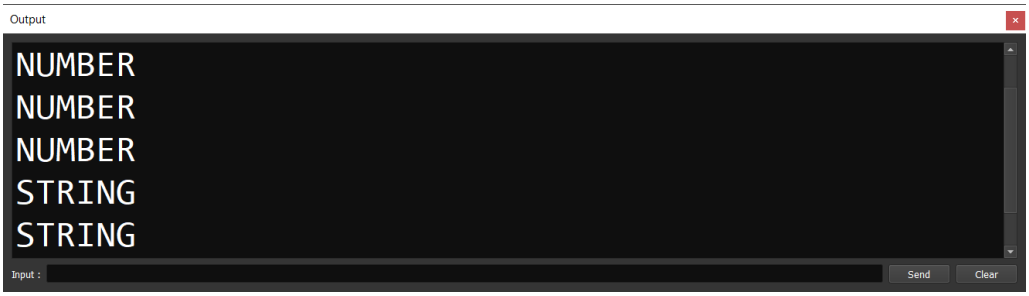
طباعة نوع البيانات – الدالة type

لاحظ! أنه فيما يلي تم اظهار نوع بيانات كل متغير باستخدام type

الكود

```
var1 = 733
var2 = 99.55
var3 = True
var4 = 'Hello'
var5 = "Hi"
see( NL +  type(var1) )
see( NL +  type(var2) )
see( NL +  type(var3) )
see( NL +  type(var4) )
see( NL +  type(var5) )
```

التنفيذ



متغير نصي متعدد الأسطر

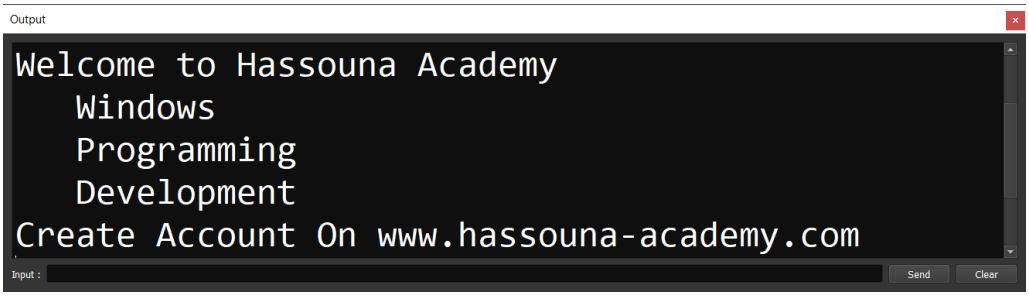
لاحظ! أنه فيما يلي تم تعريف متغير نصي يحمل اسطر نصية تطبع كما هي

الكود

```
my_str =  "
Welcome to Hassouna Academy
   Windows
   Programming
   Development
Create Account On www.hassouna-academy.com
"
```

```
see( NL + my_str)
```

التنفيذ



الكود

```
my_str =  '
Welcome to Hassouna Academy
   I love Ring
Now Create Account On www.hassouna-academy.com
'
see( NL + my_str)
```

التنفيذ

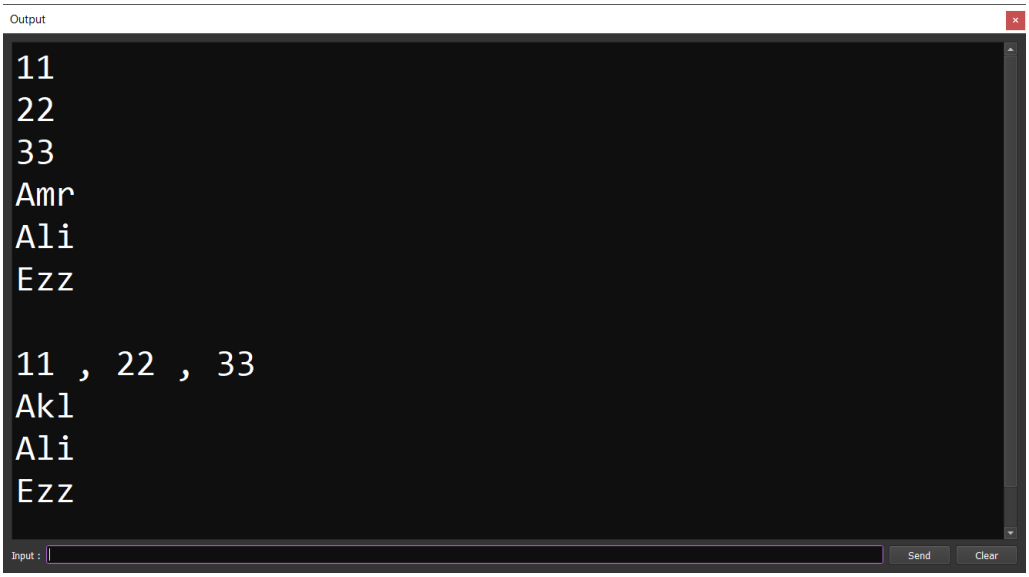
المتغير القائمة List

لاحظ! أنه فيما يلي تم استخدام متغير list قائمة بها قيم والنداء عليها بالرقم index وهو يبدأ من الـ 1، ويمكن تغيير القيم به

الكود

```
numbers = [11,22,33]
names   = ['Amr','Ali','Ezz']
see( numbers ) see( names )
see( NL +  numbers[1] )
see( " , " +  numbers[2] )
see( " , " +  numbers[3] + NL )
names[1] = 'Akl'
see( names )
```

التنفيذ



الكود

```
person = [1,'Ahmed',3500.55,True]
see( person)
```

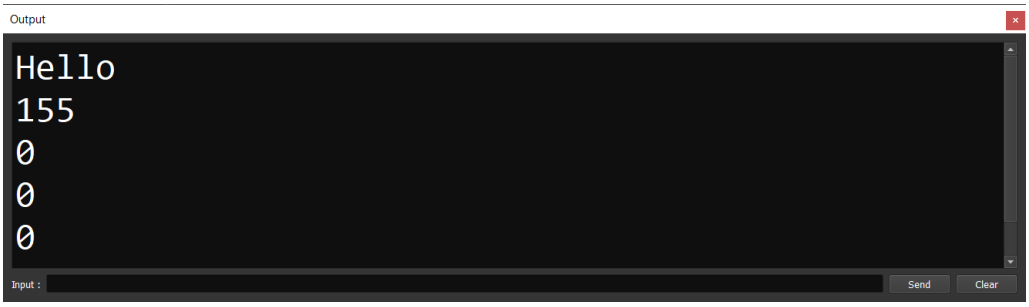
التنفيذ

لاحظ! أنه فيما يلي تم انشاء قائمة مع الحجم

الكود

```
myList = list(5)
myList[1] = "Hello"
myList[2] = 155
see myList
```

التنفيذ

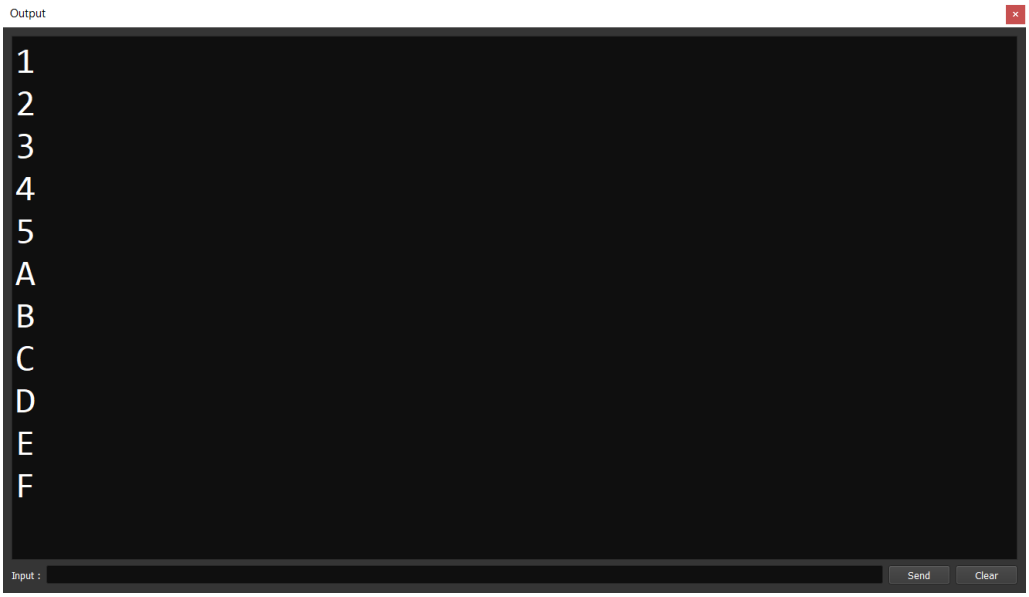


لاحظ! أنه تم عمل List بالـ Range بالرمز : بفضل الله

الكود

```
myList1 = 1:5
myList2 = 'A':'F'
see myList1
see myList2
```

التنفيذ



لاحظ! أنه فيما يلي تم استخدام add و+ لوضع قيمة واستخدام del لحذف قيمة واستخدام insert لوضع قيمة في مكان معين

الكود

```
myList = [11,"Hi"]
add( myList , 99)
myList + "OK"
insert( myList , 1 , "After item 1")
myList + 1101
del( myList , 6)
see myList
```

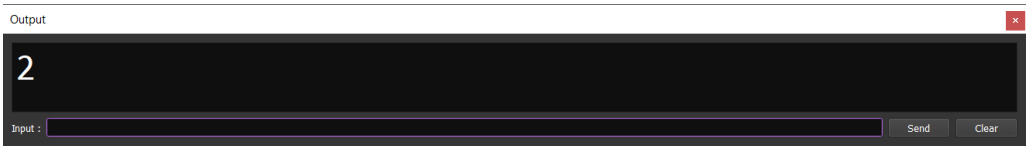
التنفيذ

لاحظ! أنه فيما يلي تم البحث عن رقم العنصر بـ find

الكود

```
myList = ["Ahmed","Amr","Adel"]
myFind = find( myList , "Amr" )
see myFind
```

التنفيذ

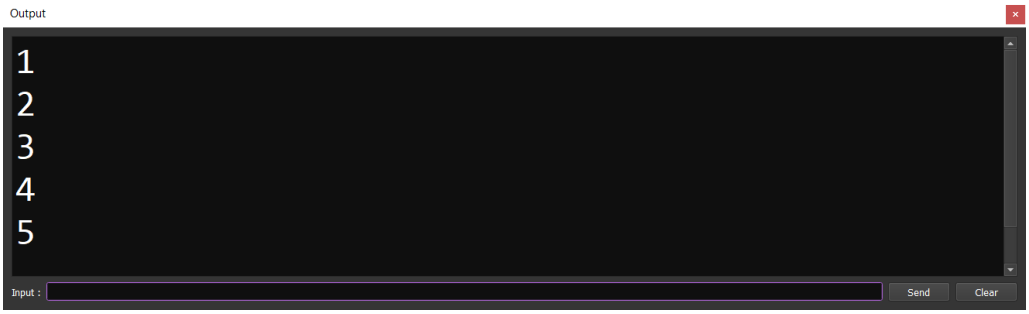


لاحظ! أنه فيما يلي تم ترتيب القائمة بـ sort ويمكن استخدامها مع الحروف والاسماء كذلك

الكود

```
myList = [3,2,5,4,1]
see sort(myList)
```

التنفيذ

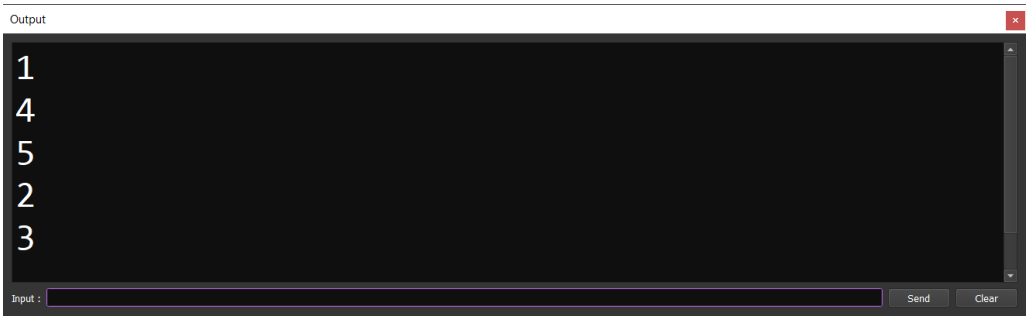


لاحظ! أنه فيما يلي تم اعكاس ما بداخل القائمة بـ reverse

الكود

```
myList = [3,2,5,4,1]
see reverse(myList)
```

التنفيذ



متغير القوائم المتداخلة

لاحظ! أنه فيما يلي تم عمل متغير يحمل بداخلة lists تتفرع منه

الكود

```
family1 = ['Ahmed','Adel','Amr']
family2 = ['Ehab','Mahmoud','Ezz']
family3 = ['Sarah','Hajer','Rehab']
```

```
family4 = ['Tawfeek','Ezzat','Foaad']
family5 = ['Abdelrahman','Abdelkareem']
family6 = ['Hasan','Shokry','Ali','Akl']

home1   = [ family1 , family2 , family3 ]
home2   = [ family4 , family5 , family6 ]

see( NL +  home1[1][2] )
see( NL +  home1[2][1] )
see( NL +  home2[1][3] )
see( NL +  home2[3][4] )
see( home1[1] )
del(home1 , 2)
see( home1 )
```

التنفيذ

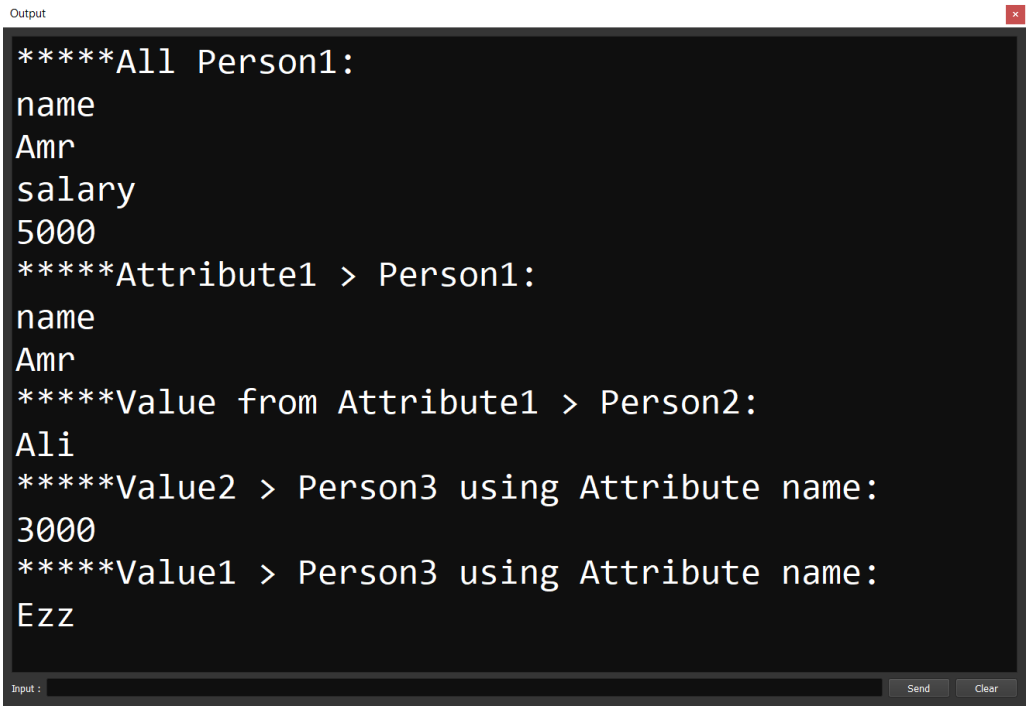
المتغير القاموس او الـ Hash Table

لاحظ! أنه فيما يلي تم استخدام متغيرات من نوع خاص وهو dictionary او hash table فيتم كتابة : ثم كتابة اسم خاصية ثم الرمز = ثم قيمتها ثم فاصلة , ثم الخاصية التي تليها مع قيمتها، وهكذا، حيث ان كل قيمة تحمل مفتاح key وقيمة value حيث يتم النداء على القيم باستخدام المفاتيح حيث يتم كتابة اسم الخاصية في شكل نصي أو يتم كتابة اسمها بكتابة : ثم كتابتها، أو يتم النداء على رقم الخاصية ثم الوصول الى اسم الخاصية بالرقم 1 او قيمتها بالرقم 2، وتم التوضيح لكل الحالات في الكود، فتستطيع بهذه الطريقة ان تنظم قيمك بالكامل بشكل منتظم

الكود

```
person1 = [:name='Amr', :salary=5000]
person2 = [:name='Ali', :salary=4000]
person3 = [:name='Ezz', :salary=3000]

see '*****All Person1:' + NL
see person1
see "*****Attribute1 > Person1:" + NL
see person1[1]
see "*****Value from Attribute1 > Person2:" + NL
see person2[1][2] + NL
see "*****Value2 > Person3 using Attribute name:" + NL
see person3['salary'] + NL
see "*****Value1 > Person3 using Attribute name:" + NL
see person3[:name]
```

التنفيذ



عدد الأشياء – الدالة len

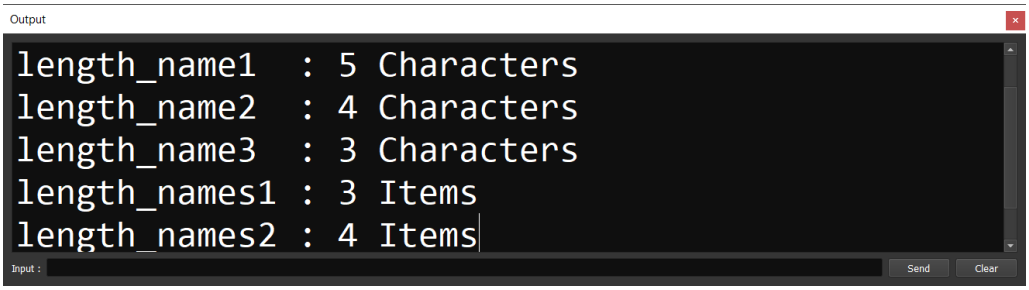
لاحظ! أنه فيما يلي تمت عملية عد لحروف النص وتم معرفة عدد العناصر

الكود

```
name1  = 'Ahmed'
name2  = 'Adel'
name3  = 'Amr'

names1 = ['Ahmed','Adel','Amr']
names2 = ['Sarah','Hajer','Rehab','Heba']

length_name1 = len(name1)
length_name2 = len(name2)
length_name3 = len(name3)
```

```
length_names1 = len(names1)
length_names2 = len(names2)

see NL +  'length_name1  : ' + length_name1 + ' Characters'
see NL +  'length_name2  : ' + length_name2 + ' Characters'
see NL +  'length_name3  : ' + length_name3 + ' Characters'

see NL +  'length_names1 : ' + length_names1 + ' Items'
see NL +  'length_names2 : ' + length_names2 + ' Items'
```

التنفيذ



التعليقات التي لا تنفذ مع الكود

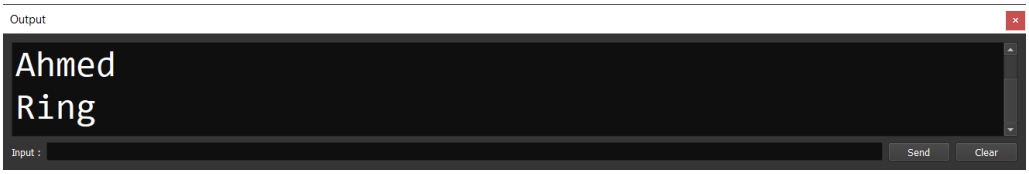
لاحظ! أنه فيما يلي تم عمل تعليقات كملاحظات ولا تنفذ مع الكود

الكود

```
#This Words Not Run, But Comment
name = 'Ahmed' #This is my name
see NL + name + NL #See name in here
//This Comment for one line
#The Hash Symbol For One Line Comment
#Can be multiline using # in each first line
/*
And Can be multiline using slash and star
In First Paragraph
```

```
And In Last Paragraph star and slash
*/
lang = "Ring" #This is my favorite language
see lang
```

التنفيذ



عوامل التخصيص

لاحظ! أنه فيما يلي تم استخدام معامل = ومعاملات تخصيص الإضافة مثل المعامل =+ أي اجعل المتغير كما هو وزد عليه، او المعامل =/ أي اجعل المتغير كما هو واقسم عليه، وهكذا

الكود

```
num = 5 see( NL + num)
num += 6 see( NL + num)
num -= 4 see( NL + num)
num *= 2 see( NL + num)
num /= 3 see( NL + num)
num %= 3 see( NL + num)
```

التنفيذ

لاحظ! أنه فيما يلي تم استخدام معامل التخصيص =+ مع النص، بحيث يحتفظ بالنص الموجود ثم يزيد عليه

الكود

```
my_str =  ""
my_str += "Hello"
my_str += " "
my_str += "Ahmed"
see( NL + my_str)
```

التنفيذ

التحويل من رقم لنص – الدالة string

لاحظ! أنه فيما يلي تم تحويل الرقم الى نص باستخدام الدالة str حتى نستطيع دمجه مع نص آخر

الكود

```
num = 99
see( NL +  'My number is: ' + string(num) )
```

التنفيذ

التحويل من نص لرقم – الدالة number

لاحظ! أنه فيما يلي تم عمل تحويل عدد صحيح int وعدد كسور float

الكود

```
num_1 = '77'
num_2 = '77.33'
num1  = number(num_1)
num2  = number(num_2)
see( NL +  num1 + ' > ' + type(num1) )
see( NL +  num2 + ' > ' + type(num2) )
```

التنفيذ

تحويل الحرف الى ASCII – الدالة ascii

لاحظ! أنه فيما يلي تم الوصول الى ascii الحرف بالدالة ord

الكود

```
c = 'A'
i = ascii(c)
see( NL +  i )
```

التنفيذ

تحويل الـ ASCII لحرف – الدالة char

لاحظ! أنه فيما يلي تم الوصول الى الحرف بوضع ascii بالدالة chr

الكود

```
i = 65
c = char(i)
see( NL +  c )
```

التنفيذ

استلام من المستخدم – الدالة give

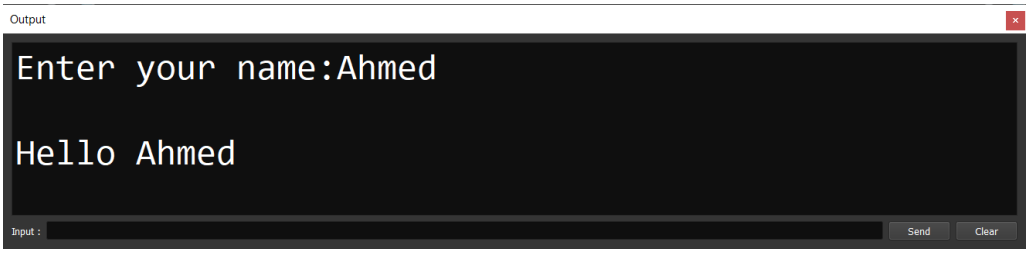
لاحظ! أنه فيما يلي انتظرك البرنامج لتدخل قيمة بالدالة give ولاحظ ان هذه القيمة نصية حتى ولو أدخلت رقم

الكود

```
see "Enter your name:"
give name
```

```
see( NL +  'Hello ' + name )
```

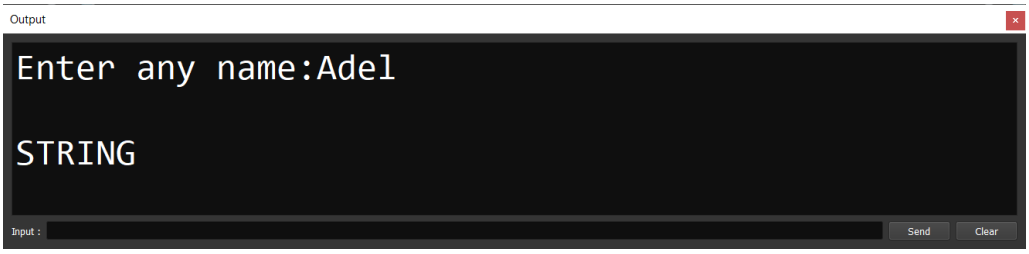
التنفيذ



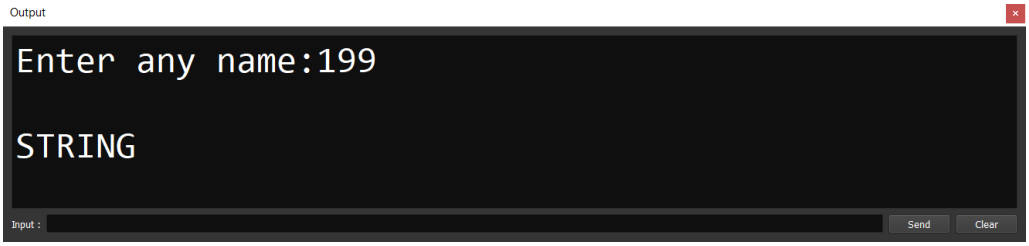
الكود

```
see 'Enter any name:' give name
see( NL +  type(name) )
```

التنفيذ 1



التنفيذ 2



لاحظ! أنه فيما يلي تم تحويل الرقم النصي باستخدام number ليكون قابل لتنفيذ العمليات الحسابية

الكود

```
see 'Enter number 1: ' give num1
see 'Enter number 2: ' give num2
result = number(num1) + number(num2)
see result
```

التنفيذ

العشوائية – الدالة random

لاحظ! أنه فيما يلي تم انشاء رقم عشوائي مباشر بالدالة random

```
see random() + NL
see random() + NL
see random()
```

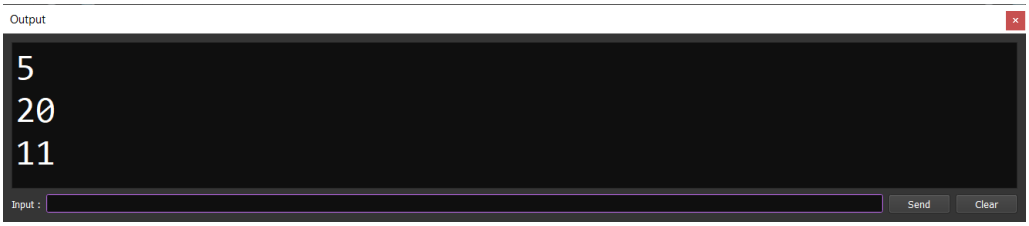
التنفيذ 1

لاحظ! أنه فيما يلي تم ضبط رقم Max بقيمة 20 أي يبدأ من الصفر ولا يتعدى 20

الكود

```
see random(20) + NL
see random(20) + NL
see random(20)
```

التنفيذ

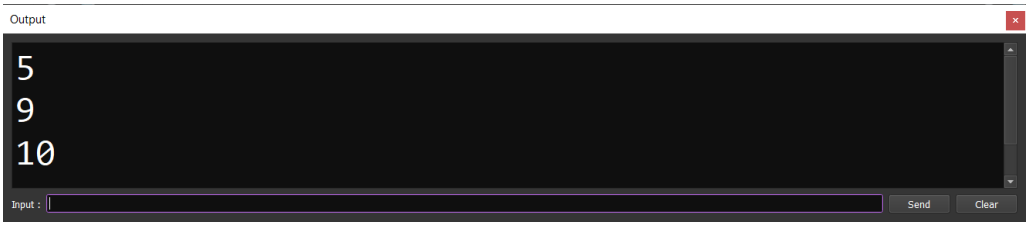


لاحظ! أنه فيما يلي تم عمل متغير min لأقل قيمة ومتغير max لأقصي قيمة ليكون المدي range للرقم العشوائي ما بين الـ min والـ max، والتفصيل في حلقة رقم 56 من كورس لغة Ring

الكود

```
min = 5
max = 10
see random(max-min) + min see NL
see random(max-min) + min see NL
see random(max-min) + min see NL
```

التنفيذ



العوامل المنطقية بالكلمات and, or, not

لاحظ! أن and لا ترجع True الا إذا كل أطرافها True

لاحظ! أن or لا ترجع False الا إذا كل أطرافها False

لاحظ! أن not تعكس القيمة من True الى False والعكس

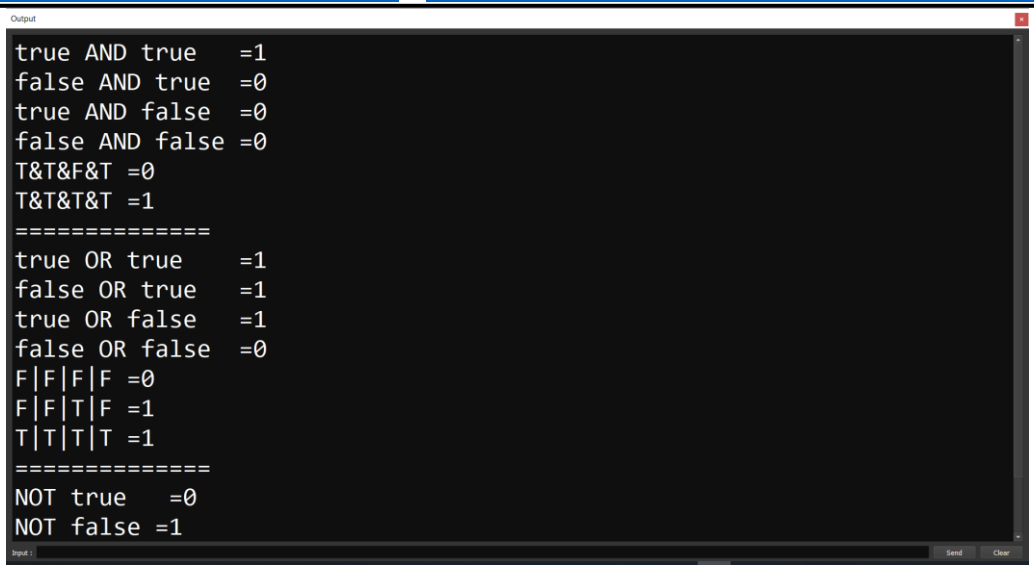
لاحظ! أن فهم القيم المنطقية لابد منه وسينعكس عليك بالسلب إذا لم تفهمه جيداً

لاحظ! أنه فيما يلي تم تفصيل الموضوع باستفاضة للمعاملات and وor وnot لأهميتهم في عالم البرمجة

الكود

```
bool1 = True and True    see 'true AND true   ='+bool1+NL
bool2 = False and True   see 'false AND true  ='+bool2+NL
bool3 = True and False   see 'true AND false  ='+bool3+NL
bool4 = False and False  see 'false AND false ='+bool4+NL
bool5 = True and True and False and True  see 'T&T&F&T ='+bool5+NL
bool6 = True and True and True and True   see 'T&T&T&T ='+bool6+NL
see '==============='+NL
bool7  = True or True    see 'true OR true    ='+bool7+NL
bool8  = False or True   see 'false OR true   ='+bool8+NL
bool9  = True or False   see 'true OR false   ='+bool9+NL
bool10 = False or False  see 'false OR false  ='+bool10+NL
bool11 = False or False or False or False see 'F|F|F|F ='+bool11+NL
bool12 = False or False or True or False  see 'F|F|T|F ='+bool12+NL
bool13 = True or True or True or True     see 'T|T|T|T ='+bool13+NL
see '==============='+NL
bool14 = not True see 'NOT true   ='+bool14+NL
bool15 = not False see 'NOT false ='+bool15+NL
```

التنفيذ



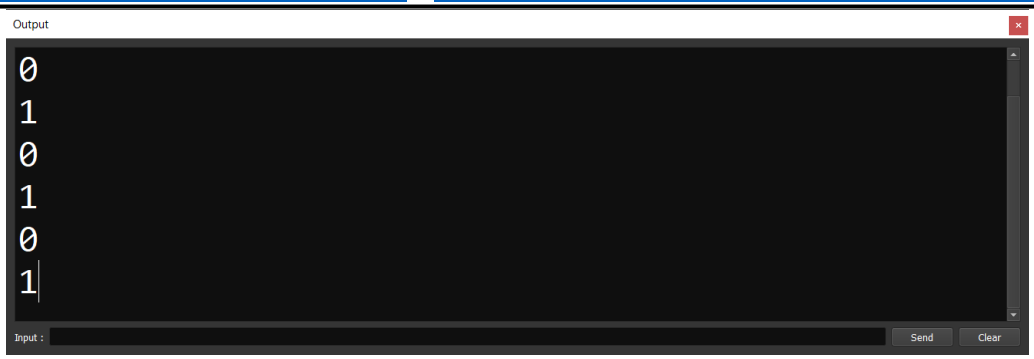
عوامل المقارنة

لاحظ! أنه فيما يلي تم استخدام عوامل المقارنة لاختبار شرط معين
ولاحظ ان 1 تعتبر True و0 تعتبر False

الكود

```
x =7 y = 9
b1 = (x>y) see( NL + b1)
b2 = (x<y) see( NL + b2)
b3 = (x>=y) see( NL + b3)
b4 = (x<=y) see( NL + b4)
b5 = (x=y) see( NL + b5)
b6 = (x!=y) see( NL + b6)
```

التنفيذ



تحويلات النظم العددية

لاحظ! أنه فيما يلي تم التحويل ما بين النظم العددية بأكثر من طريقة

الكود

```
num_d = 255
num_h = 'ff'
see 'Hexadecimal   :' + hex(num_d) + NL
see 'Decimal from h:' + dec(num_h) + NL
see str2hex('Hello') + NL
see hex2str('48656c6c6f') + NL
```

التنفيذ

تحميل المكتبات - load

لاحظ! أنه فيما يلي تم تحميل المكتبة القياسية للغة الرينج وهي بالملف stdlib.ring ثم تم التعامل مع الدالة print الموجودة بها للطباعة بدون see مع \n لنزول سطر بدلاً من NL

الكود

```
load "stdlib.ring"
print("Hello\n\n")
```

التنفيذ

تقطيع النص – الدالة substr

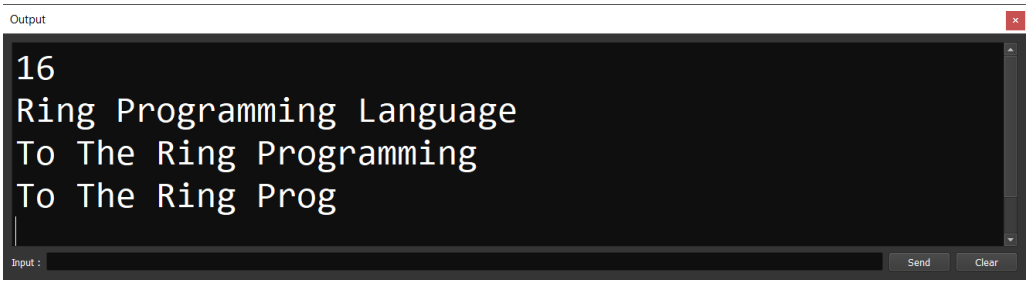
لاحظ! أنه فيما يلي تم تقطيع النص عن طريق وضع نص ونبحث عن كلمة فنأتي برقم الحرف ثم نضعه ليكون بداية التقطيع، أو يمكن ان نضع البداية والنهاية لقطع نص معين

الكود

```
str = "Welcome To The Ring Programming Language"
index = substr(   str    ,   "Ring"  )
str2 = substr(  str    ,    index    )
str3 = substr(  str   ,   9    ,   23   )
str_start = substr(str   ,   "To")
str_end   = substr(str   ,   "Ring")
str4 = substr( str , str_start , str_end )
see index + NL
```

```
see str2 + NL
see str3 + NL
see str4 + NL
```

التنفيذ



الكود

```
text = "A Welcome To Ring> Programming Language Home Page Z"
s = substr( text   ,   "Ring"  )
e = substr( text ,  " Home" )
newText = substr( text  ,  s  ,   e-s  )
see newText
```

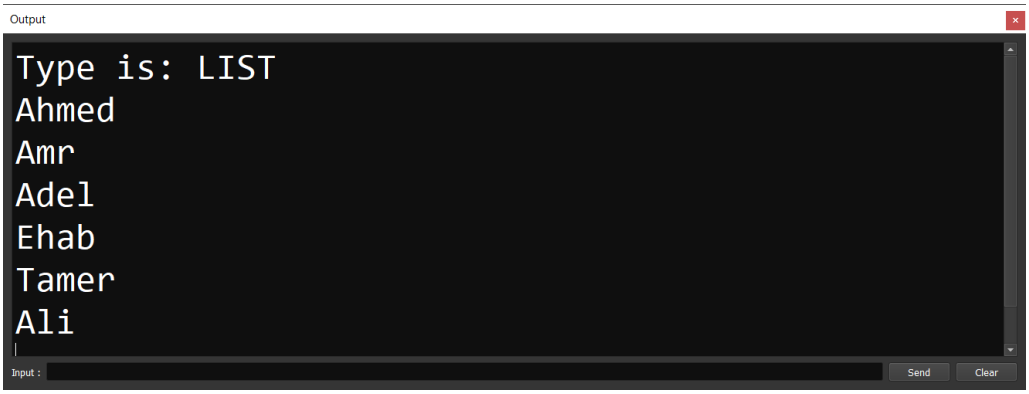
التنفيذ

لاحظ! أنه فيما يلي تم استخدام الدالة split لفصل النص الى قائمة

الكود

```
load "stdlib.ring"
names = "Ahmed Amr Adel Ehab Tamer Ali"
myNames = split(names , " ")
print( 'Type is: ' + type(myNames)+"\n" )
see myNames
```

التنفيذ



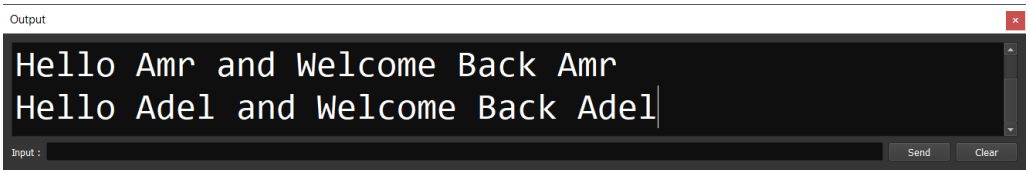
استبدال النص – الدالة substr

لاحظ! أنه فيما يلي تم استبدال نص بنص آخر

الكود

```
my1 = 'Hello Amr and Welcome Back Amr'
my2 = substr(my1 , 'Amr' , 'Adel')
see( NL +  my1 )
see( NL +  my2 )
```

التنفيذ



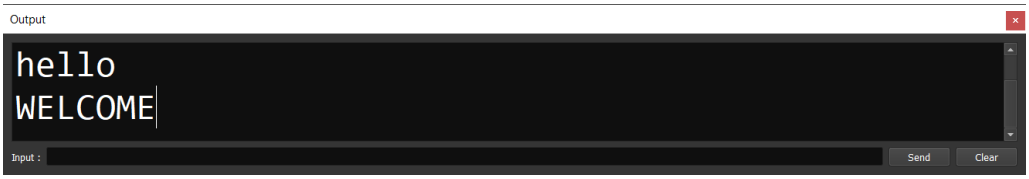
تحولات النص الكبير والصغير ABC, abc

لاحظ! أنه فيما يلي تم تحويل النص upper وlower

الكود

```
str1 = 'HELLO'
str2 = 'welcome'
see( NL +  lower(str1) )
see( NL +  upper(str2) )
```

التنفيذ



تحقق النص

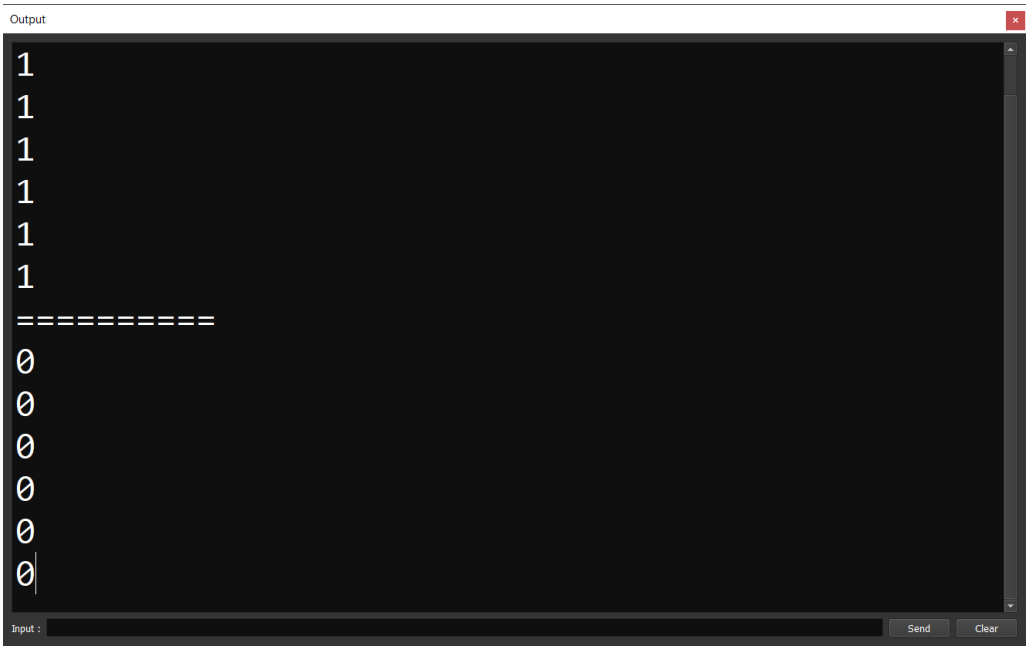
لاحظ! أنه فيما يلي تم التحقق من النص لمعرفة هل هو أحرف كبيرة او صغيرة او ارقام او حروف او مسافات او ما الى ذلك، ولاحظ أنه لو كان التحقق بـ نعم فإن الناتج هو True وهو 1 ولو كان بـ لا فإن الناتج هو False وهو صفر

الكود

```
see( NL +  isupper('HELLO') )
see( NL +  islower('hello') )
see( NL +  isalpha('HEllo') )
see( NL +  isalnum('ABC45') )
see( NL +  isdigit('12345') )
see( NL +  isspace('     ') )
see( NL + '==========')
see( NL +  isupper('HeLLO') )
see( NL +  islower('HeLLO') )
see( NL +  isalpha('HE7lo') )
see( NL +  isalnum('AB@45') )
```

```
see( NL +  isdigit('12A45') )
see( NL +  isspace('   .   ') )
```

التنفيذ



اتخاذ القرار – جملة if

لاحظ! أنه فيما يلي تم تنفيذ الكود متوقف على شرط معين فإذا كان ناتج الشرط بـ True أي 1 ينفذ وإذا كان بـ False أي صفر لا ينفذ الكود

```
x = 5
if x=5
   see( NL + 'OK')
ok
```

التنفيذ

الكود

```
x = 7
if x>5 see 'OK1 ' see 'OK2 ' see 'OK3' ok
```

التنفيذ

الكود

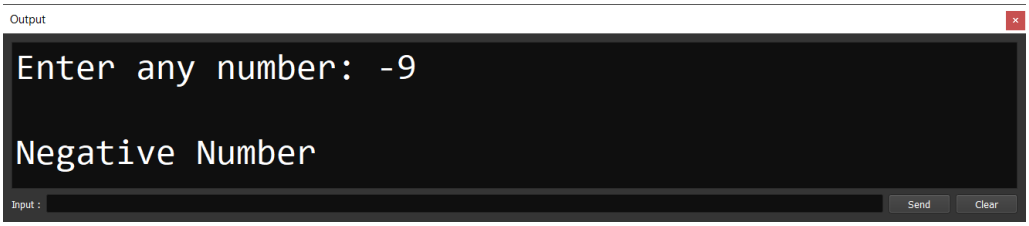
```
x = 7
if x<=7
    see NL + 'OK1'
    see NL + 'OK2'
    see NL + 'OK3'
ok
```

التنفيذ

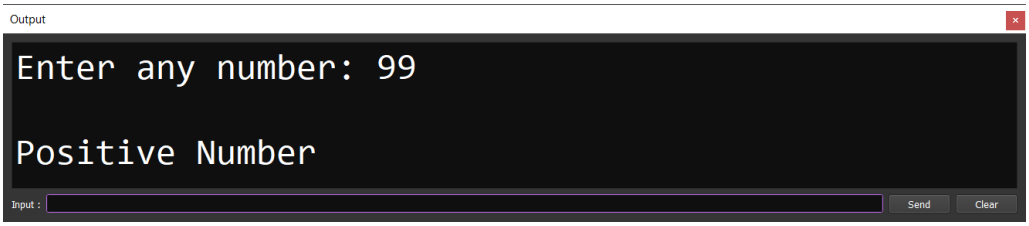
الكود

```
see 'Enter any number: ' give num
if number(num)<0
    see( NL + 'Negative Number')
else
    see( NL + 'Positive Number')
ok
```

التنفيذ 1



التنفيذ 2

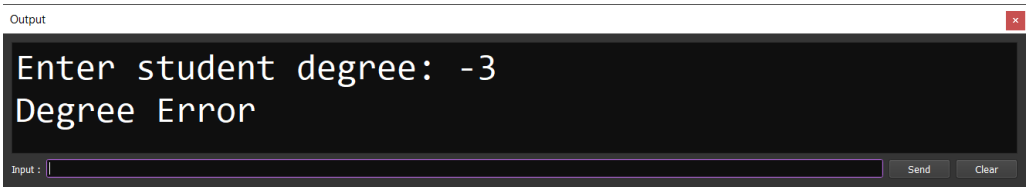


الكود

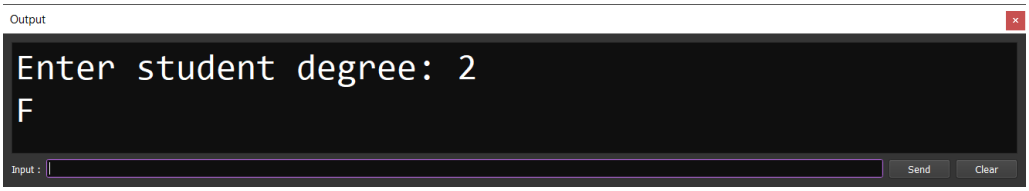
```
see 'Enter student degree: ' give _degree
degree = number(_degree)
if degree<0 or degree>100
    see 'Degree Error'
but degree<50
    see 'F'
but degree<60
    see 'E'
but degree<70
    see 'D'
but degree<80
    see 'C'
```

```
but degree<90
    see 'B'
else
    see 'A'
ok
```

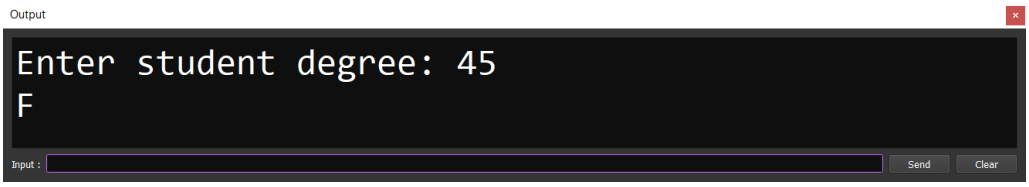
التنفيذ 1



التنفيذ 2

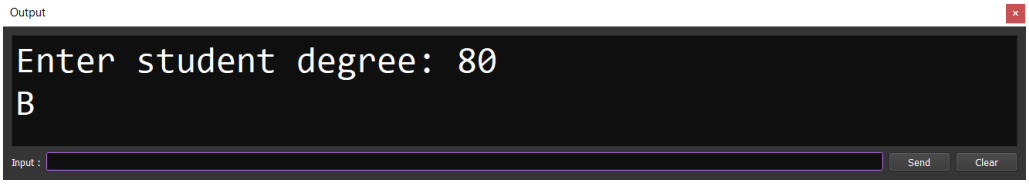


التنفيذ 3

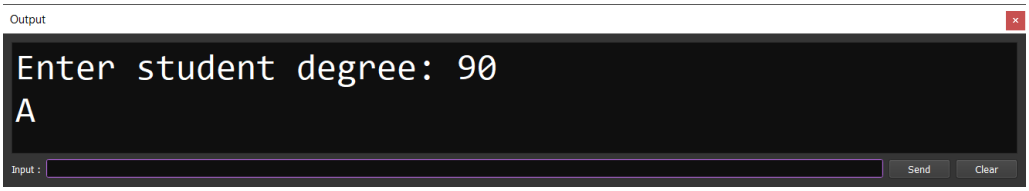


التنفيذ 4

التنفيذ 5



التنفيذ 6



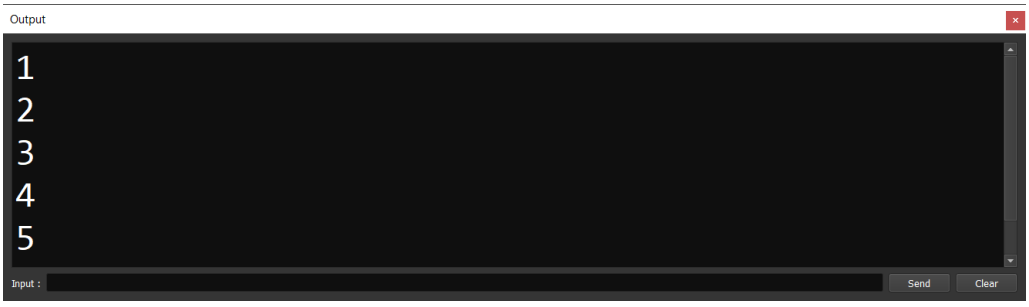
حلقات التكرار – الجملة for

لاحظ! أنه فيما يلي تم استخدام حلقات التكرار لاستخدام تكرار الكود وعمل عمليات متعددة بأسطر كود قليلة لتوفير الوقت والجهد

الكود

```
for x in [1,2,3,4,5] see x + NL next
```

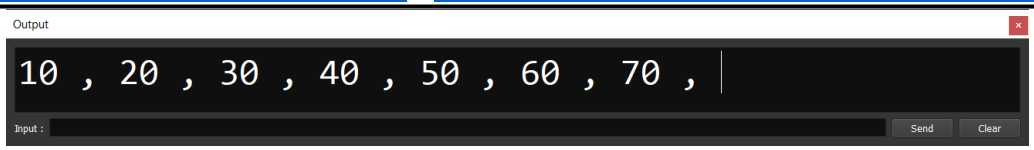
التنفيذ



الكود

```
for x in [10,20,30,40,50,60,70]
   see string(x) + " , "
next
```

التنفيذ



الكود

```
for x=1 to 7
   see string(x) + " , "
next
```

التنفيذ

الكود

```
for x=2 to 11 step 2
    see string(x) + ' , '
next
```

التنفيذ

الكود

```
for x=1 to 6
   if x!=4
      see string(x) + ' , '
   ok
next
```

التنفيذ

الكود

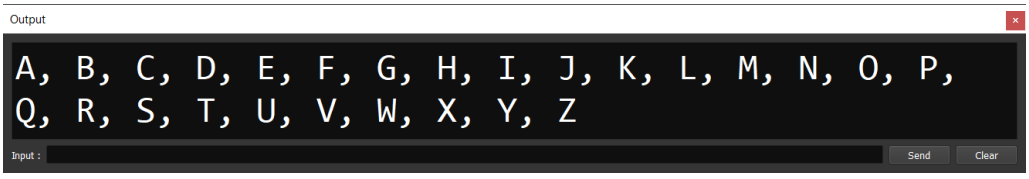
```
for x=5 to 1 step -1
    see string(x) + ' , '
next
```

التنفيذ

الكود

```
alpha = ''
for x=ascii('A') to ascii('Z')
    alpha += char(x)
    if x<ascii('Z') alpha += ', ' ok
next
see  alpha
```

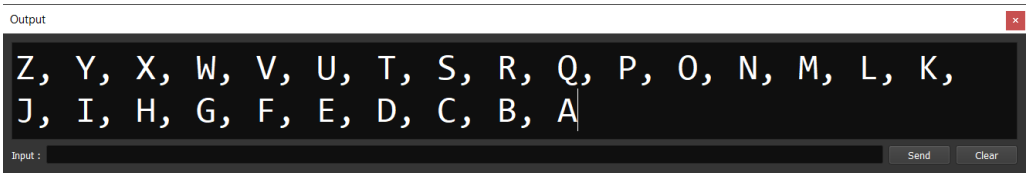
التنفيذ



الكود

```
alpha = ''
for x=ascii('Z') to ascii('A') step -1
    alpha += char(x)
    if x>ascii('A') alpha += ', ' ok
next
see  alpha
```

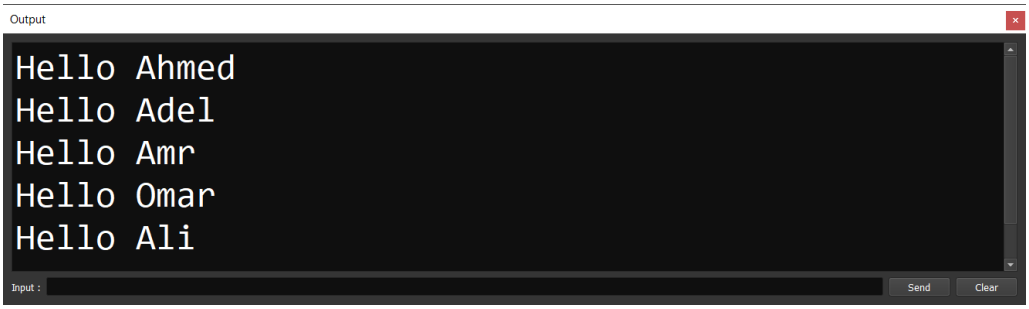
التنفيذ



الكود

```
names = ['Ahmed','Adel','Amr','Omar','Ali']
for x=1 to len(names)
    see 'Hello ' + names[x] + NL
next
```

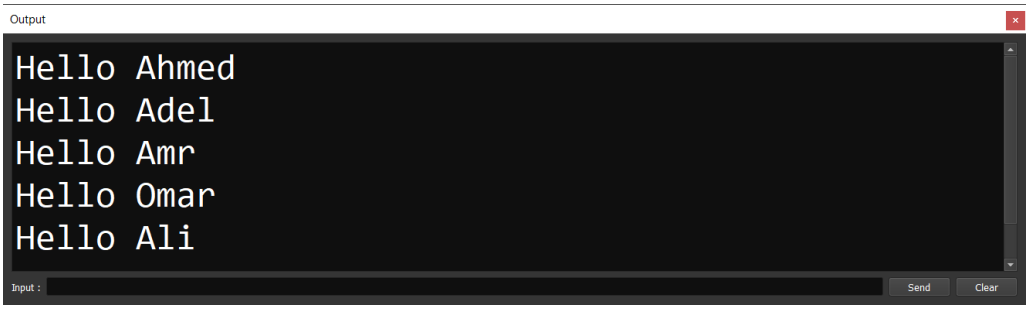
التنفيذ



الكود

```
names = ['Ahmed','Adel','Amr','Omar','Ali']
for x in names
    see 'Hello ' + x + NL
next
```

التنفيذ



الكود

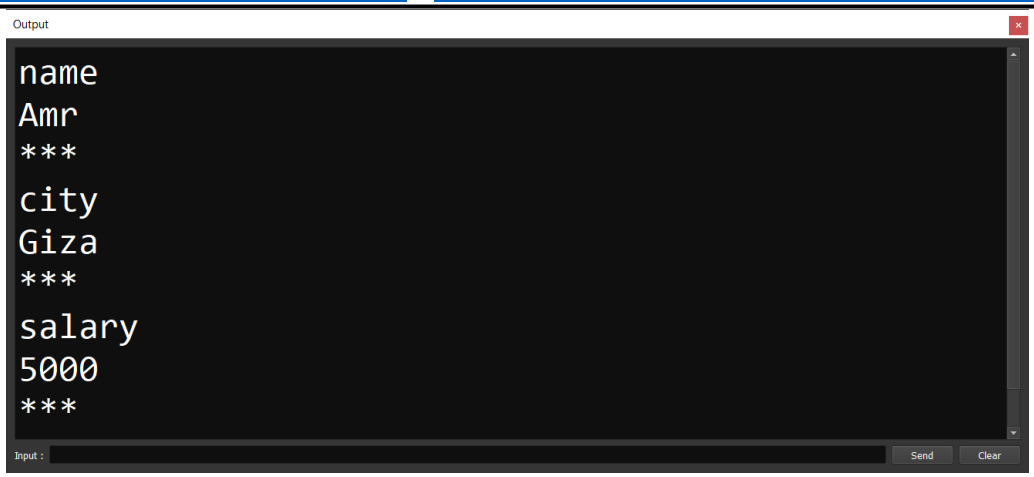
```
my_list = [ 3, 'A', True, 5.7 ]
for v in my_list
    see type(v) + NL
next
```

التنفيذ

الكود

```
emp = [:name='Amr', :city='Giza', :salary=5000]
for x in emp
   see x + '***'
next
```

التنفيذ



الكود

```
emp = [:name='Amr', :city='Giza', :salary=5000]
for x in emp
   see x[1] + '***'
next
```

التنفيذ

الكود

```
emp = [:name='Amr', :city='Giza', :salary=5000]
for x in emp
   see x[2] + '***'
next
```

التنفيذ

الكود

```
emp1 = [:name='Amr', :city='Giza', :salary=5000]
emp2 = [:name='Ali', :city='Alex', :salary=6000]
emp3 = [:name='Ezz', :city='Sina', :salary=7000]
emps = [ emp1 , emp2 , emp3 ]
for x in emps
   see x[:name] + '***' + x['salary'] + NL
next
```

التنفيذ

التكرار المتداخل nested loop

لاحظ! أنه فيما يلي تم استخدام التكرار المتداخل والذي يحتاج منك قمة التركيز والفهم ليكون بسيط بالنسبة لك، ولاحظ ان التكرار المتداخل يمكن ان يكون loop داخل loop أو أكثر من ذلك

الكود

```
family1 = ['Ahmed','Adel','Amr']
family2 = ['Ehab','Mahmoud','Ezz']
family3 = ['Sarah','Hajer','Rehab']
```

```
home1   = [ family1 , family2 , family3 ]

for x=1 to len(home1)
    see 'Family: ' + x + NL
    for y=1 to len(home1[x])
        see '   Name' + y + ' is: ' + home1[x][y] + NL
    next
next
```

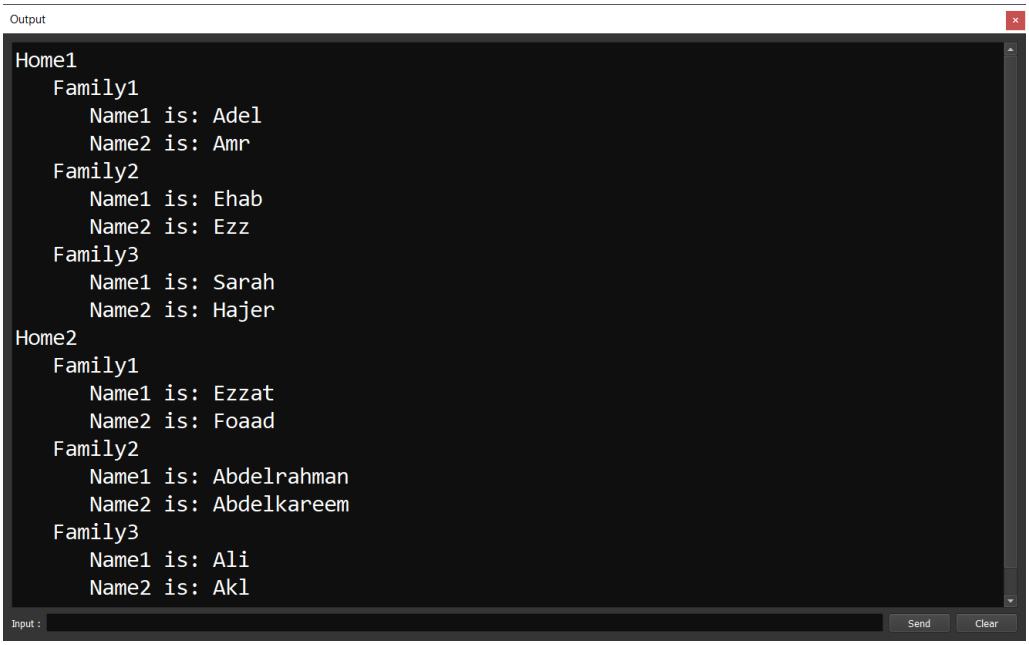
التنفيذ

الكود

```
family1 = ['Adel' , 'Amr']
family2 = ['Ehab' , 'Ezz']
family3 = ['Sarah' , 'Hajer']

family4 = ['Ezzat' , 'Foaad']
family5 = ['Abdelrahman' , 'Abdelkareem']
family6 = ['Ali' , 'Akl']
```

```
home1   = [ family1 , family2 , family3 ]
home2   = [ family4 , family5 , family6 ]

homes = [ home1 , home2 ]

for x=1 to len(homes)
    see 'Home' + x + NL
    for i=1 to len(homes[x])
        see '   Family' + i + NL
        for y=1 to len(homes[x][i])
            see '      Name' + y + ' is: ' + homes[x][i][y]
+ NL
        next
    next
next
```

التنفيذ



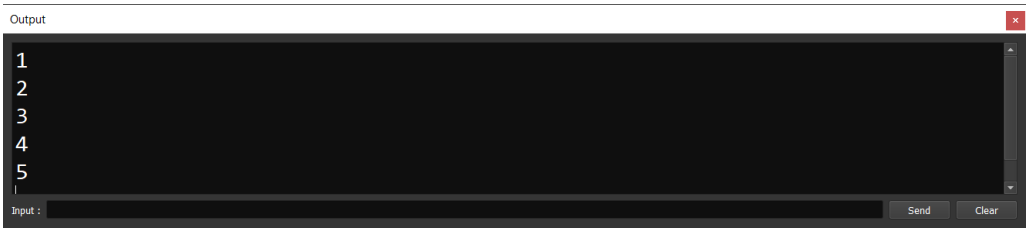
حلقات التكرار – الجملة while

لاحظ! أنه فيما يلي تم استخدام حلقات التكرار لاستخدام تكرار الكود وعمل عمليات متعددة بأسطر كود قليلة لتوفير الوقت والجهد

الكود

```
x = 1
while x <= 5
    see x + NL
    x +=1
end
```

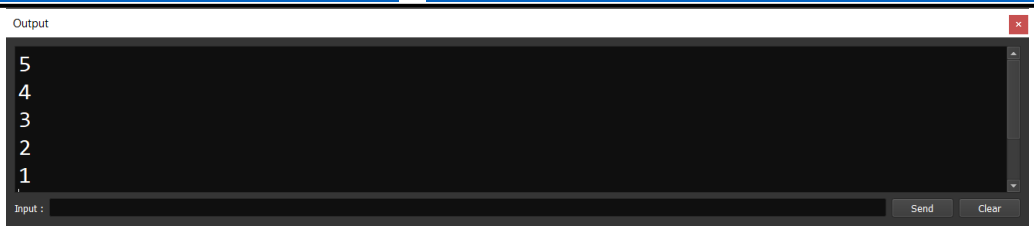
التنفيذ



الكود

```
x = 5
while x >= 1
    see x + NL
    x -= 1
end
```

التنفيذ



الكود

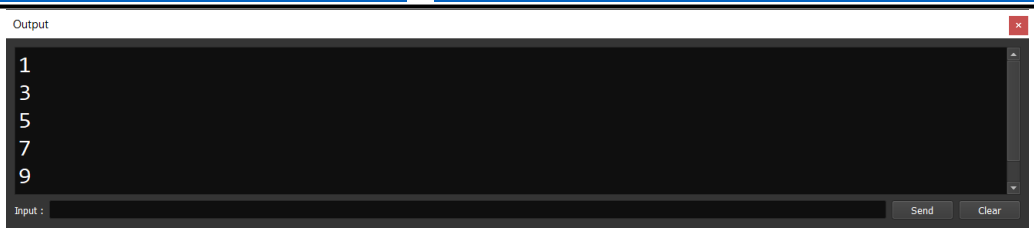
```
x = 2
while x <= 10
    see x + NL
    x += 2
end
```

التنفيذ

الكود

```
x = 1
while x <= 10
    see x + NL
    x += 2
end
```

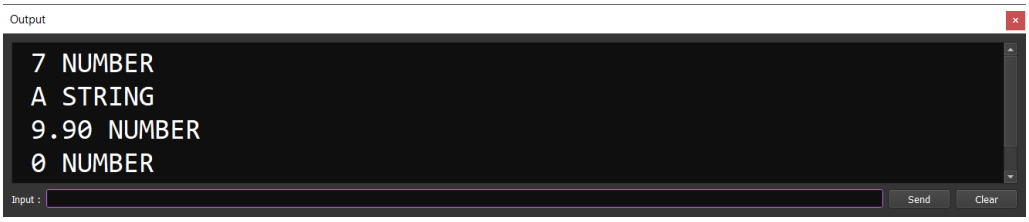
التنفيذ



الكود

```
my_list = [7,'A',9.9,False]
x = 1
while x <= len(my_list)
    see ' ' + my_list[x] + ' ' + type(my_list[x]) + NL
    x += 1
end
```

التنفيذ



حلقات التكرار – الجملة do... again

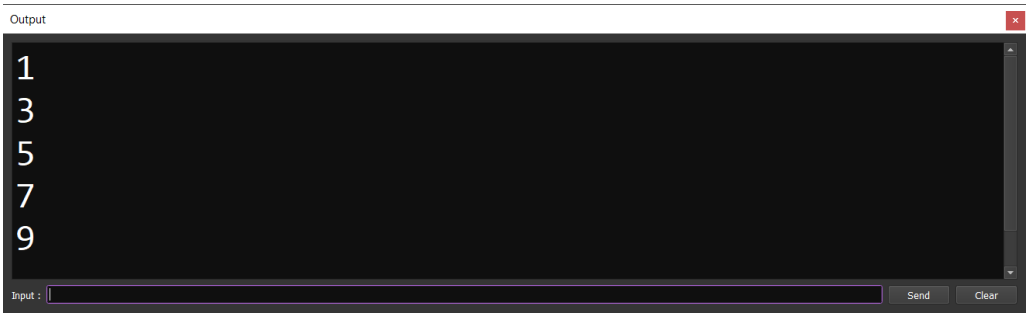
لاحظ! أنه فيما يلي تم استخدام الجملة do وفي نهايتها again ثم الشرط، وهي تختلف عن الجملة while بحيث لو أن الشرط لا يتحقق تتنفذ مرة على الأقل في البداية

الكود

```
x = 1
do
    see x + NL
```

```
    x += 2
again x <= 10
```

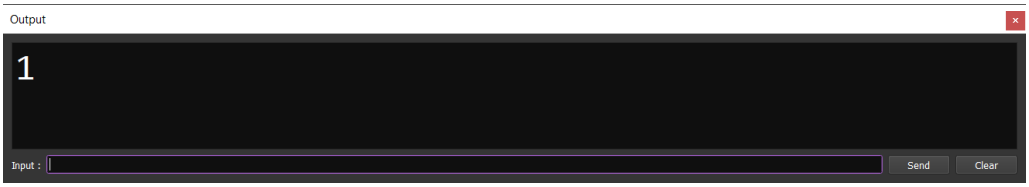
التنفيذ



الكود

```
x = 1
do
    see x + NL
    x += 2
again False
```

التنفيذ



التكرار اللانهائي

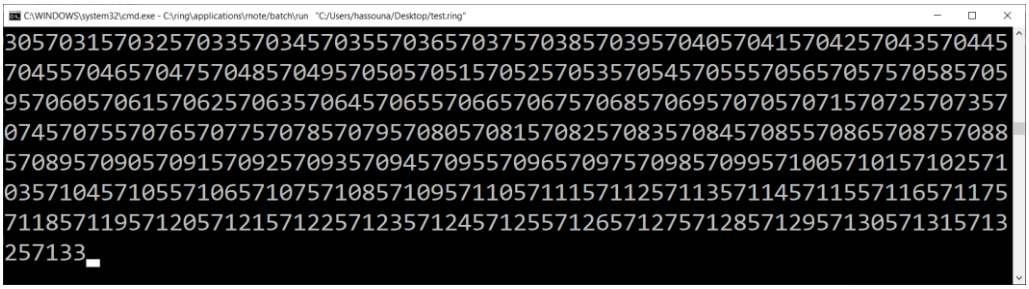
لاحظ! أن التكرار اللانهائي يجعل البرنامج لا يقف ويظل يعمل

الكود

```
x = 1
while True
```

```
    see x + ', '
    x += 1
end
```

التنفيذ



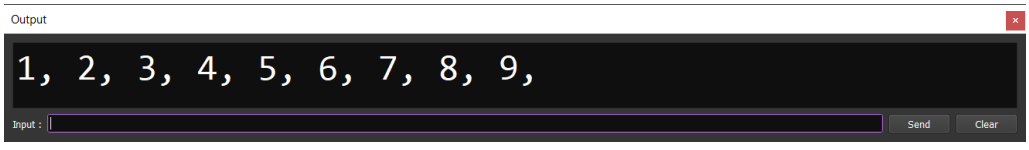
الخروج من التكرار - exit

لاحظ! أنه فيما يلي تم الخروج من التكرار نهائياً ووقفة تماماً باستخدام الجملة exit

الكود

```
x = 1
while True
    see string(x) + ', '
    x += 1
    if x >= 10
        exit
    ok
end
```

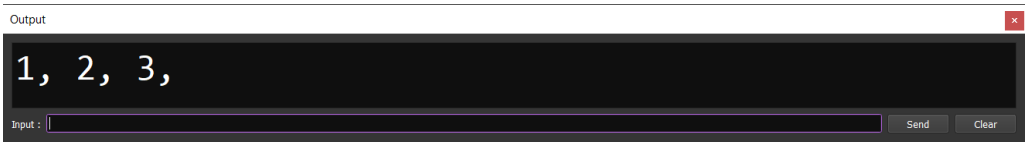
التنفيذ



الكود

```
x = 1
while x <= 10
   see string(x) + ', '
   x += 1
   if x >= 4   exit   ok
end
```

التنفيذ



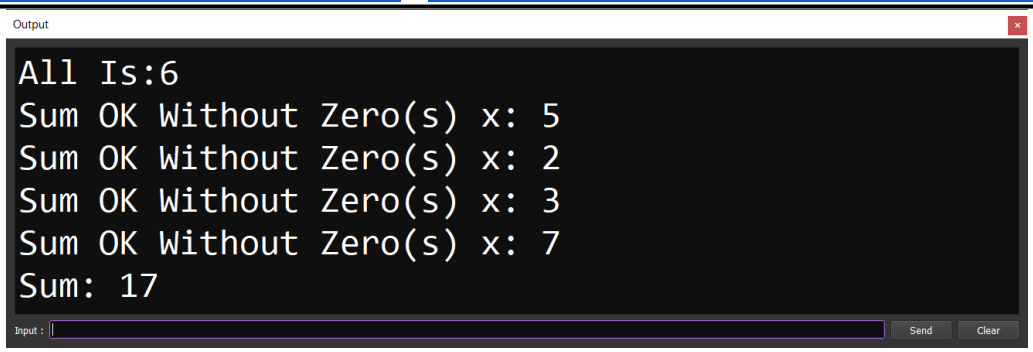
الاستكمال في التكرارات - loop

لاحظ! أن الـ loop تتجاهل ما تحتها وتذهب لتستمر بشكل طبيعي في التكرار

الكود

```
numbers = [5,2,0,3,0,7]
mysum = 0
see 'All Is:' + len(numbers) + NL
for x in numbers
    if x = 0
       loop
    ok
    mysum += x
    see 'Sum OK Without Zero(s) ' + 'x: ' + x + NL
next
see 'Sum: ' + mysum
```

التنفيذ



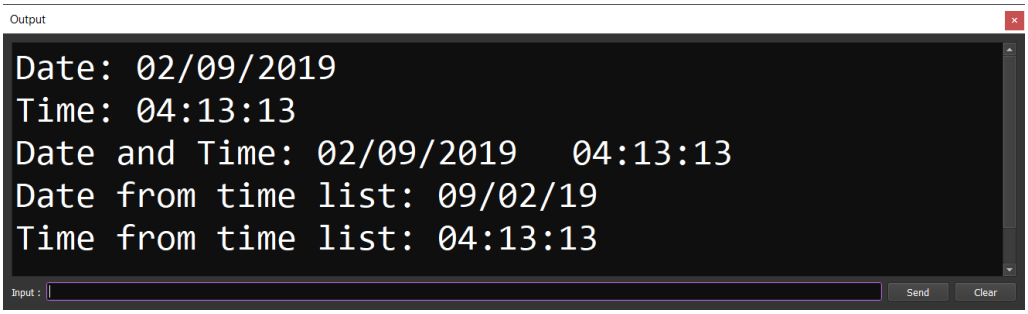
الوقت والتاريخ – date, time, timelist

لاحظ! أنه تم التعامل مع الوقت والتاريخ بسهولة

الكود

```
see "Date: " + date() + NL
see "Time: " + time() + NL
see "Date and Time: " + date() + TAB + time() + NL
see "Date from time list: " +  timelist()[16] + NL
see "Time from time list: " +  timelist()[17] + NL
```

التنفيذ

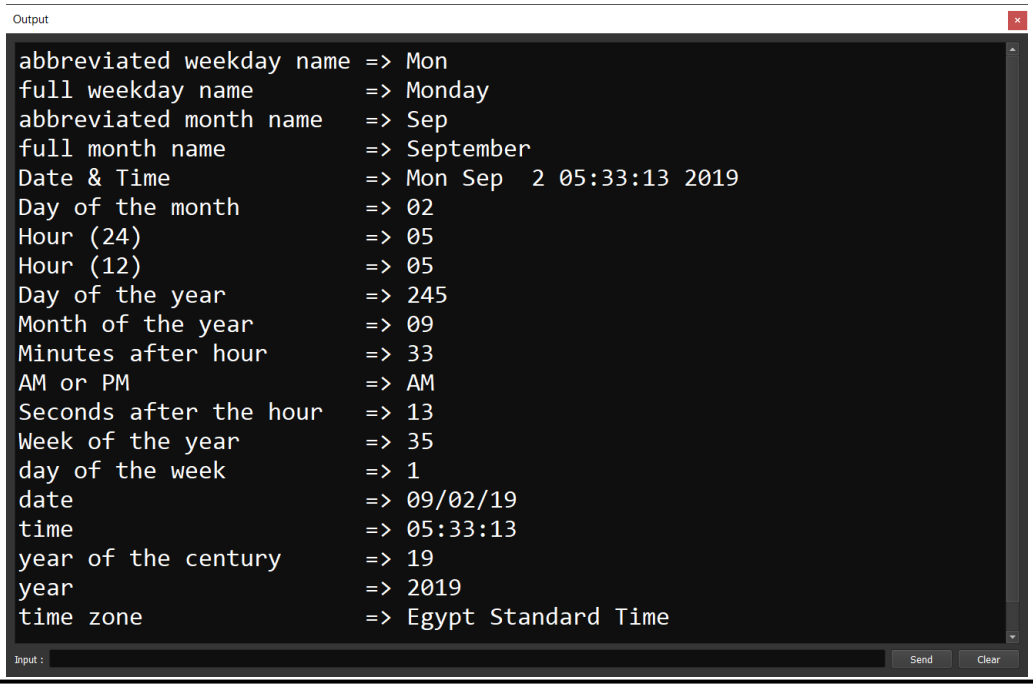


الكود

```
see 'abbreviated weekday name => ' + timelist()[1]  + NL
see 'full weekday name        => ' + timelist()[2]  + NL
```

```
see 'abbreviated month name   => ' + timelist()[3]  + NL
see 'full month name          => ' + timelist()[4]  + NL
see 'Date & Time              => ' + timelist()[5]  + NL
see 'Day of the month         => ' + timelist()[6]  + NL
see 'Hour (24)                => ' + timelist()[7]  + NL
see 'Hour (12)                => ' + timelist()[8]  + NL
see 'Day of the year          => ' + timelist()[9]  + NL
see 'Month of the year        => ' + timelist()[10] + NL
see 'Minutes after hour       => ' + timelist()[11] + NL
see 'AM or PM                 => ' + timelist()[12] + NL
see 'Seconds after the hour   => ' + timelist()[13] + NL
see 'Week of the year         => ' + timelist()[14] + NL
see 'day of the week          => ' + timelist()[15] + NL
see 'date                     => ' + timelist()[16] + NL
see 'time                     => ' + timelist()[17] + NL
see 'year of the century      => ' + timelist()[18] + NL
see 'year                     => ' + timelist()[19] + NL
see 'time zone                => ' + timelist()[20] + NL
```

التنفيذ



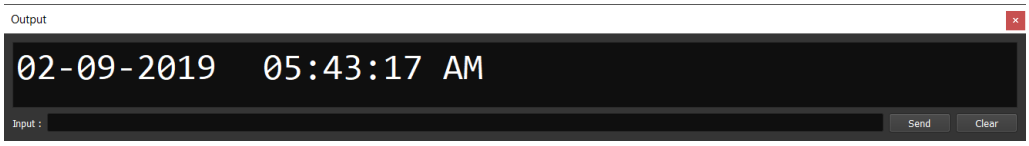
تخصيص التاريخ والوقت

لاحظ! أنه فيما يلي تم تخصيص الوقت والتاريخ كما نريد

الكود

```
TL = timelist()
myTime = TL[8] + ':' + TL[11] + ':' + TL[13] + ' ' + TL[12]
myDate = TL[6] + '-' + TL[10] + '-' + TL[19]
see myDate + TAB + myTime
```

التنفيذ

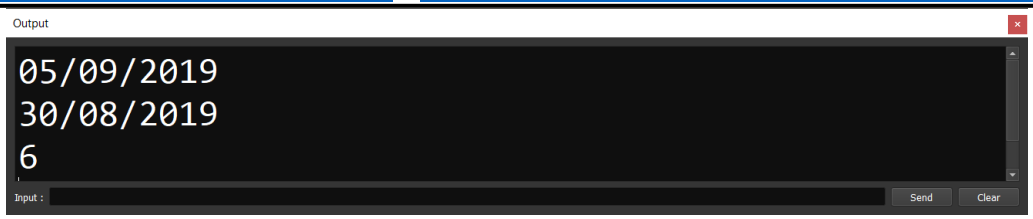


التلاعب مع التاريخ والوقت

الكود

```
d1 = adddays( date()   ,  3  )
d2 = adddays(  date()  ,  -3)
see d1 + nl
see d2 + nl
diffD = diffdays(  d1   ,   d2)
see diffD + nl
```

التنفيذ

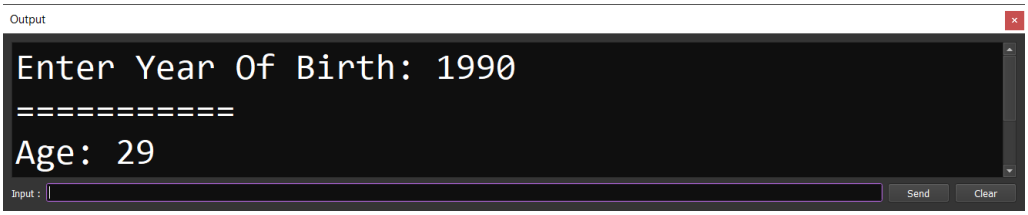


برنامج حساب العمر

الكود

```
see "Enter Year Of Birth: "
give by

cy = timelist()[19]

age = cy - by

see "===========" + nl
see "Age: " + age + nl
see "===========" + nl
```

التنفيذ



فتح ملف نصي موجود لتجهيزه للقراءة

لاحظ! أنه فيما يلي تم استخدام fopen لفتح ملف موجود مع النمط r الذي يستخدم للقراءة ولاحظ انه يتم غلق الملف بالدالة fclose بعد استخدامه

الكود

```
file = fopen( 'my_file.txt' , 'r' )
fclose(file)
```

التنفيذ

يتم فتح الملف واغلاقه، ولو كان غير موجود يحدث خطأ

انشاء ملف نصي فارغ وتجهيزه للكتابة

لاحظ! أنه فيما يلي تم استخدام w ومعني ذلك ان الملف سيتم إنشاؤه، ولو كان موجود سيتم حذفه ثم انشاء ملف جديد، ولاحظ أنه لابد من اغلاق الملف بالدالة fclose بعد استخدامه

الكود

```
file = fopen( 'my_file.txt' , 'w' )
fclose(file)
```

التنفيذ

يتم انشاء الملف في نفس مكان ملف الأكواد فقط ثم اغلاق الملف

انشاء ملف نصي والكتابة عليه

لاحظ! أنه تم إضافة حروف بالملف باستخدام الدالة fputc

الكود

```
file = fopen( 'my_file.txt' , 'w' )
fputc( file , 'H' )
fputc( file , 'E' )
fputc( file , 'L' )
fputc( file , 'L' )
fputc( file , 'O' )
fclose(file)
```

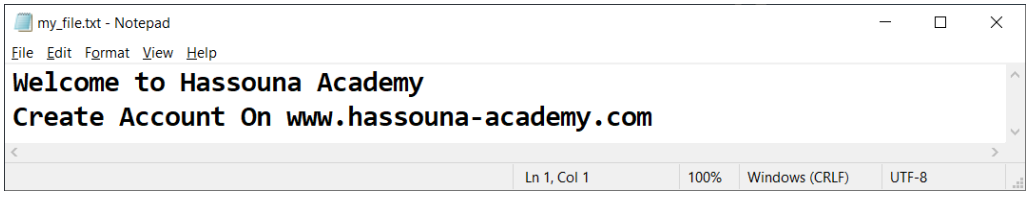
التنفيذ

لاحظ! أنه تم إضافة نصوص بالملف باستخدام الدالة fputs

الكود

```
file = fopen( 'my_file.txt' , 'w' )
fputs( file , 'Welcome to Hassouna Academy' + NL )
fputs( file , 'Create Account On www.hassouna-academy.com' )
fclose(file)
```

التنفيذ

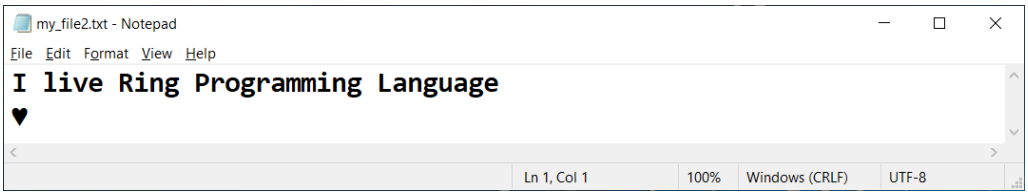


لاحظ! أنه تم إضافة نصوص بالملف باستخدام الدالة fwrite

الكود

```
file = fopen( 'my_file2.txt' , 'w' )
text = 'I live Ring Programming Language' + NL
text += '♥'
fwrite( file , text )
fclose(file)
```

التنفيذ



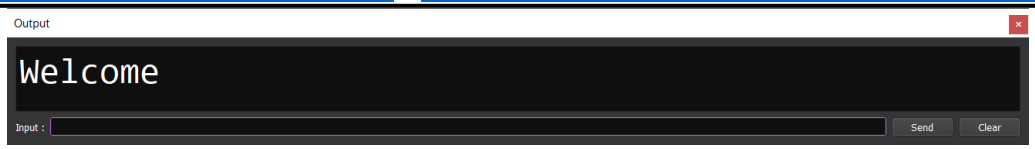
القراءة من ملف موجود

لاحظ! أنه تم قراءة حرف تلو الحرف من الملف باستخدام الدالة fgetc

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text  = fgetc(file)
text += fgetc(file)
text += fgetc(file)
text += fgetc(file)
text += fgetc(file)
text += fgetc(file)
text += fgetc(file)
fclose(file)
see text
```

التنفيذ

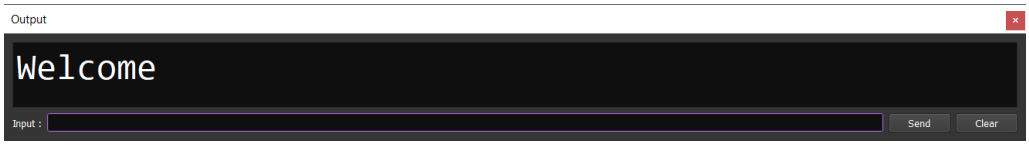


لاحظ! أنه تم قراءة حروف مجمعة من الملف باستخدام الدالة fgets ويتم تحديد عدد الحروف التي نريد قراءتها، ولكن اذا تجاوزت السطر لا تأتي بالسطر التالي

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text  = fgets( file , 7 )
fclose(file)
see text
```

التنفيذ

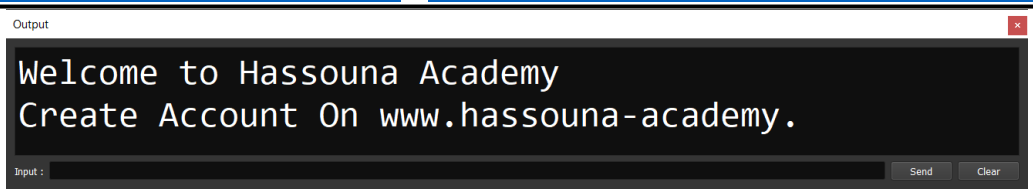


لاحظ! أنه تم قراءة حروف مجمعة من الملف باستخدام الدالة fread ويتم تحديد عدد الحروف التي نريد قراءتها، وإذا تجاوزت السطر تأتي بالسطر التالي بسهولة

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text  = fread( file , 67 )
fclose(file)
see text
```

التنفيذ



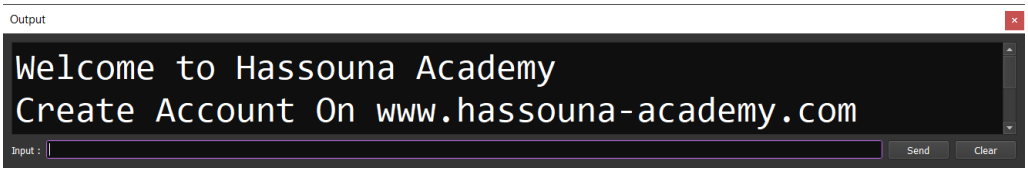
قراءة الملف بالكامل بالتكرار بأكثر من طريقة

لاحظ! أنه فيما يلي تم استخدام الدالة feof والتي تختبر الوصول الى نهاية الملف حيث تكون بـ True لو هي النهاية وتكون بـ False إذ لم تكن النهاية ثم تم نفيها بـ not حيث وكأننا نقول لتكرار الـ while قم بالتكرار طالما لم تصل الى نهاية الملف

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text = ''
while not feof(file)
   c = fgetc(file)
   if c != -1
      text += c
   ok
end
fclose(file)
see text
```

التنفيذ

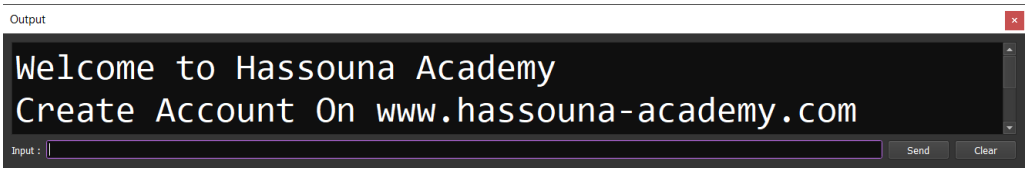


لاحظ! أنه فيما يلي تم استخدام isString لاختبار وجود النص

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text = ''
c    = ''
do
   text += c
   c     = fgetc(file)
again isString(c)
fclose(file)
see text
```

التنفيذ

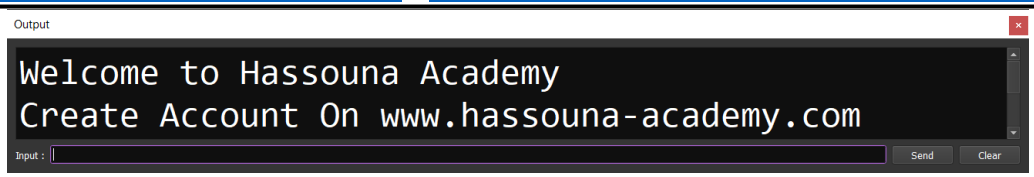


لاحظ! أنه فيما يلي تم قراءة 50 حرف لكل لفة تكرار ويمكن تغيرها

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text = ''
c    = ''
while isString(c)
   text += c
   c     = fgets(file,50)
end
fclose(file)
see text
```

التنفيذ



الكتابة على الملف بسهولة – الدالة write

الكود

```
text = 'Welcome to Hassouna Academy' + NL
text += 'Create Account On www.hassouna-academy.com'
write( 'my_file.txt' , text )
```

التنفيذ

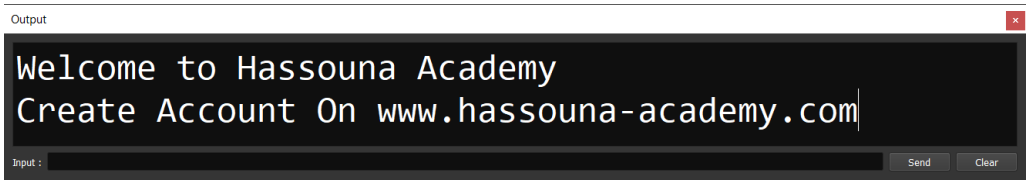
سوف يتم انشاء ملف وكتابة نص بداخلة

القراءة من الملف بسهولة – الدالة read

الكود

```
text = read('my_file.txt')
see text
```

التنفيذ



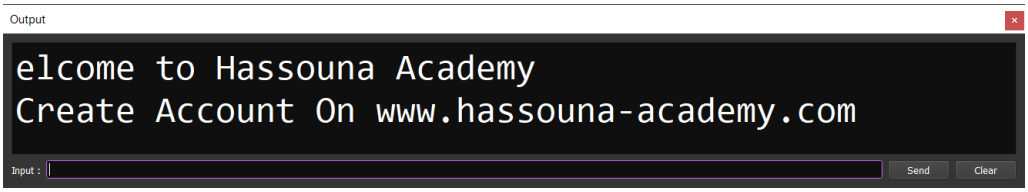
القراءة من ملف من مكان معين

لاحظ! أنه تم استخدام seek لإيقاف مؤشر القراءة في مكان معين قبل القراءة، ولاحظ أن 2 للوصول الى نهاية الملف وصفر للبداية

الكود

```
file = fopen( 'my_file.txt' , 'r' )
text = ''
C_FILESTART = 0
C_FILEEND = 2
fseek( file , 1 , C_FILESTART )
while not feof(file)
   c = fgetc(file)
   if c != -1
      text += c
   ok
end
fclose(file)
see text
```

التنفيذ



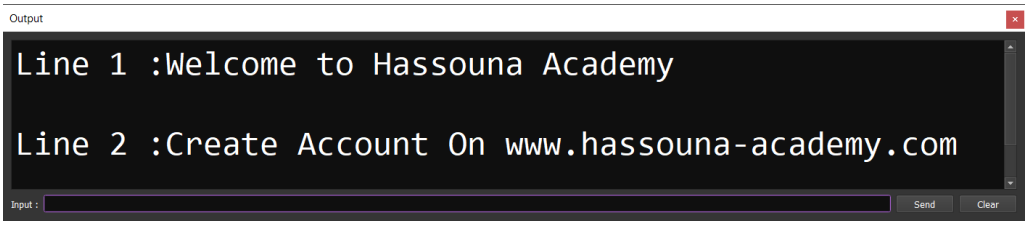
القراءة من ملف سطر تلو السطر

لاحظ! أنه فيما يلي تم استخدام readline والتي تبدأ بقراءة اول سطر ثم التالي ثم الذي يليه وهكذا

الكود

```
Load "stdlib.ring"
fp = fopen("my_file.txt","r")
linecount = 0
while not feof(fp)
linecount += 1
see 'Line ' + linecount +' :'+ Readline(fp) + NL
end
fclose(fp)
```

التنفيذ



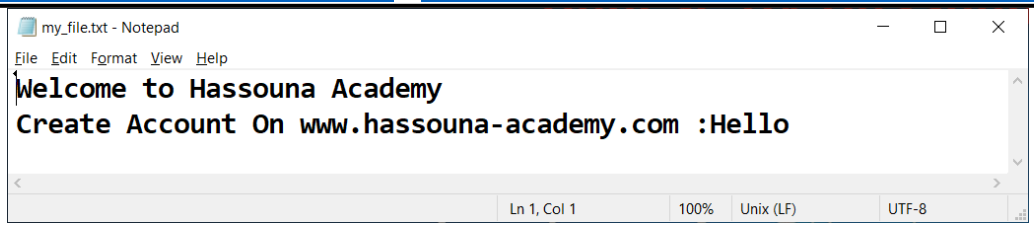
تكملة الكتابة على ملف نصي موجود

لاحظ! أنه فيما يلي تم استخدام a لعمل Append وتكملة الكتابة على الملف دون مساس الكلام القديم

الكود

```
fp = fopen("my_file.txt","a")

fputs( fp , ' :Hello' )

fclose(fp)
```

التنفيذ



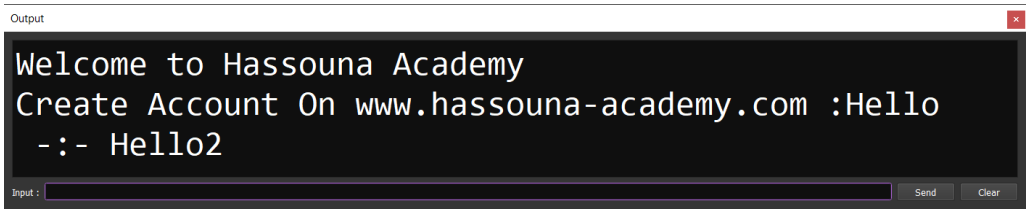
تكملة الكتابة مع القراءة من ملف نصي

لاحظ! أنه فيما يلي تم استخدام a+ للتكملة ومعها يمكن القراءة الكود

لاحظ! أنه فيما يلي تم استخدام rewind لاستفاقة الملف ليتضمن ما تم كتابته أولاً ثم ليقرأ بشكل صحيح، و:اننا نقول له استيقظ وانظر ما حدث فيك

```
file = fopen("my_file.txt","a+")
fputs( file , NL + ' -:- Hello2' )
rewind(file)
text = ''
c    = ''
do
   text += c
   c = fgetc(file)
again isstring(c)
fclose(file)
see text
```

التنفيذ



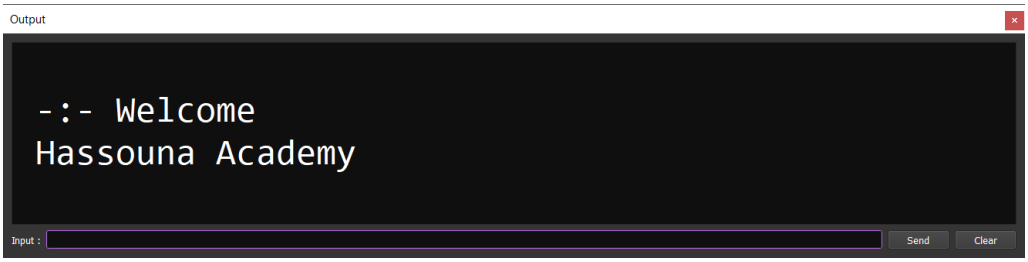
القراءة مع الكتابة لملف نصي

لاحظ! أنه فيما يلي تم استخدام w+ للكتابة ومعها يمكن القراءة

الكود

```
file = fopen("my_file3.txt","w+")
fputs( file , NL + ' -:- Welcome' )
fputs( file , NL + ' Hassouna Academy' )
rewind(file)
text = ''
c    = ''
do
   text += c
   c = fgetc(file)
again isstring(c)
fclose(file)
see text
```

التنفيذ



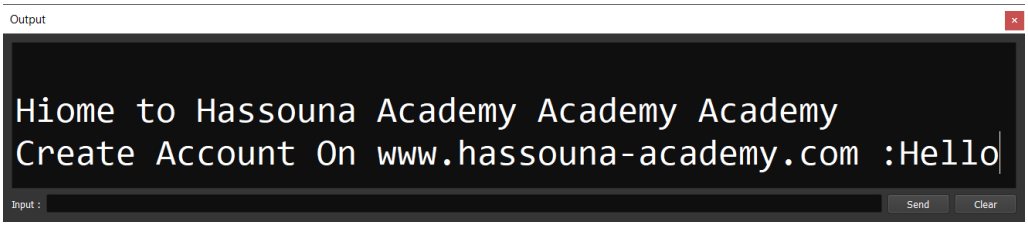
الكتابة مع القراءة

لاحظ! أنه فيما يلي تم استخدام r+ للقراءة ومعها يمكن الكتابة

الكود

```
file = fopen("my_file.txt","r+")
fputs( file , NL + 'Hi' )
rewind(file)
text = ''
c    = ''
do
   text += c
   c = fgetc(file)
again isstring(c)
fclose(file)
see text
```

التنفيذ



القراءة والكتابة والتكملة لملف binary

لاحظ! أنه فيما يلي تم استخدام نفس الأنماط السابقة مثل w و r و r+ وغيرها ولكن مع زيادة حرف b بعدها للـ Binary

الكود

```
f = fopen( 'my_file.jpg' , 'wb' )
f = fopen( 'my_file.jpg' , 'rb' )
f = fopen( 'my_file.jpg' , 'wb+' )
f = fopen( 'my_file.jpg' , 'rb+' )
f = fopen( 'my_file.jpg' , 'ab' )
f = fopen( 'my_file.jpg' , 'ab+' )
fclose(f)
```

التنفيذ

الكود هنا للمعرفة، والتعامل معه بعد ذلك حسب الطلب لملف الـBinary

التحقق من وجود ملف او مجلد وانشاء المجلدات

الكود

```
load "stdlib.ring"
if fExists("test.ring")
   see "My Ring File Here" + NL
ok
if not dirExists("My New Folder")
   OsCreateOpenFolder(' "My New Folder" ')
   see "Yes" + NL
ok
OsCreateOpenFolder('Folder1 Folder2 Folder3')
```

التنفيذ

سيتم التحقق من ملف الكود وانشاء المجلد اذ لم يكن موجود ثم المجلدات

حذف الملفات والمجلدات

الكود

```
load "stdlib.ring"
if fExists("test.txt")
   OsDeleteFile("test.txt")
ok
if dirExists("My New Folder")
   OsDeleteFolder(' "My New Folder" ')
ok
```

التنفيذ

سيتم حذف المجلد والملف إذا كانوا موجودين

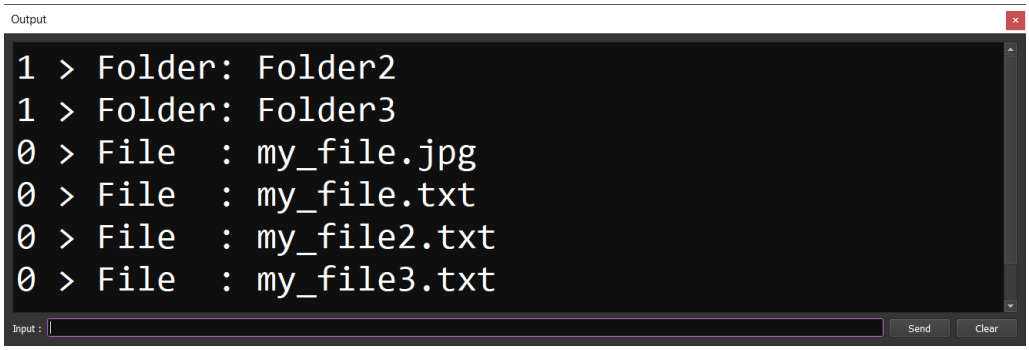
معرفة الملفات والمجلدات الموجودة بمجلد معين

لاحظ! أن الدالة dir تأتي بجميع الملفات والمجلدات للمسار المحدد لها داخل الاقواس، ولاحظ أن للمجلد 1 وللملف صفر لنعرف الملف من المجلد

الكود

```
myList = Dir("Folder1")

for item in myList
   see item[2]
   if item[2] see " > Folder: " + item[1] + nl
   else see " > File  : " + item[1] + nl ok
next
```

التنفيذ



تنفيذ أوامر الدوس

الكود

```
system('mkdir my_folder_using_dos')
```

التنفيذ

سيتم انشاء مجلد باستخدام نظام DOS، أو أي امر آخر حسب الاحتياج.

نسخ الملفات - copy

الكود

```
load "stdlib.ring"
OsCopyFile( 'Folder1/my_file.jpg   "my file.jpg" ' )
```

التنفيذ

سيتم نسخ الملف من المكان الأول الى المكان الثاني

نسخ المجلدات

الكود

```
load "stdlib.ring"
OsCopyFolder( 'Folder1/F2 ' , ' "my fol 2" ' )
```

التنفيذ

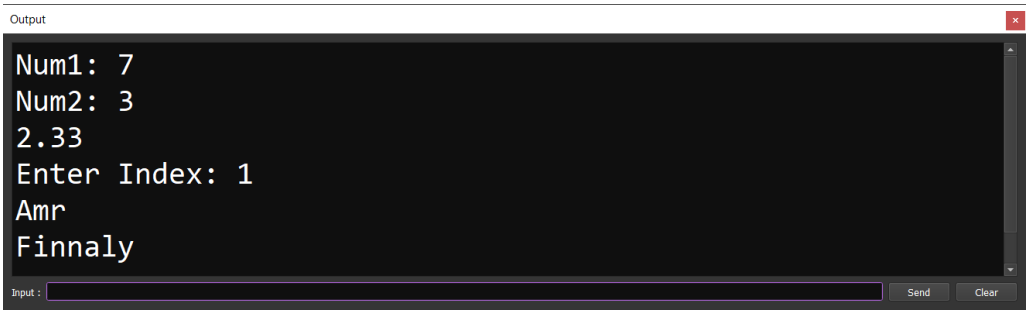
سيتم نسخ المجلد من المكان الأول الى المكان الثاني

معالجة الأخطاء

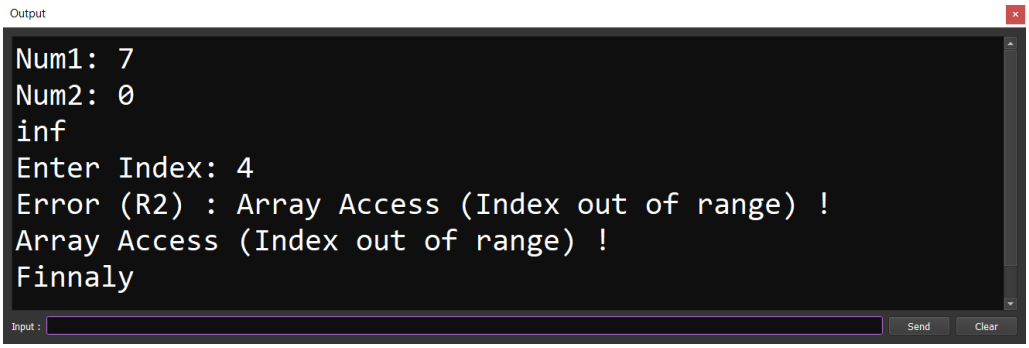
الكود

```
try
   see 'Num1: ' give num1
   see 'Num2: ' give num2
   r = num1 / num2
   see r + nl
   names = ['Amr','Ali','Ezz']
   see 'Enter Index: ' give i
   see names[ number(i) ]
catch
   see cCatchError + NL
   see substr(cCatchError, 14)
end #or_done

see NL + "Finnaly" + NL
```

التنفيذ 1



التنفيذ 2



ترتيب الكود في لغة البرمجة Ring – هام جداً

لاحظ أن التعامل مع الكود في لغة ring حتى الآن هو أوامر أو أسطر برمجية فقط وهي الـ statements، وتعاملنا مع تحميل المكتبات وهي الـ load ولكن سنعمل مع أشياء أخرى وهي الدوال functions والحزم packages والفئات classes وهنا نحتاج أن نفهم ترتيب هذه الأشياء في لغة الرينج لمراعاتها أثناء العمل، ولغة الرينج لها ترتيب لابد أن تسير عليه وهو كما يلي:

1. المكتبات **load**
2. الجمل **statements**
3. الدوال **functions**
4. الحزم والفئات **packages & classes**

لابد ان يكون هذا الترتيب في ذهنك دائماً وأنت تعمل بلغة البرمجة رينج

انشاء دوال functions عادية لا تنفذ شيء

لاحظ! أنه تم تعريف ثلاث دوال ليس بهم اكواد

الكود

```
func my_func1()
    //This is my function 1
func my_func2()
    //This is my function 2
func my_func3()
    //This is my function 3
```

التنفيذ

لا شيء سينفذ

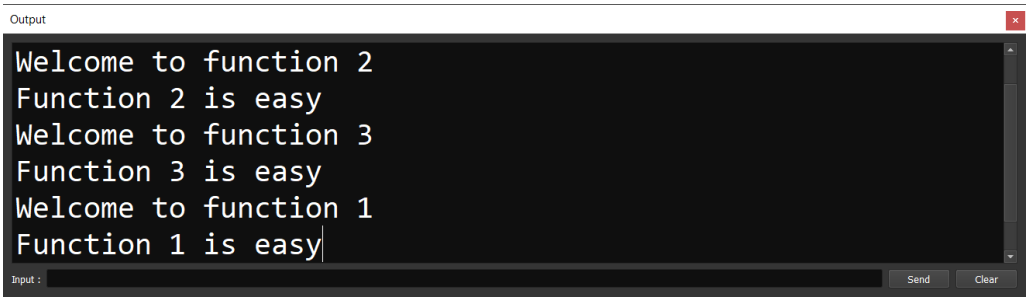
انشاء دوال بها اكواد ثم النداء عليها في الأعلى

الكود

```
my_func2()
my_func3()
my_func1()

func my_func1()
    //This is my function 1
    see( NL + 'Welcome to function 1')
    see( NL + 'Function 1 is easy')

func my_func2()
    //This is my function 1
    see( NL + 'Welcome to function 2')
    see( NL + 'Function 2 is easy')
```

```
func my_func3()
    see( NL + 'Welcome to function 3')
    see( NL + 'Function 3 is easy')
```

التنفيذ



انشاء دالة تستقبل وسائط

لاحظ! أنه فيما يلي تم عمل دالة باسم say_hello ولكن هذه الدالة تستقبل متغير باسم name ثم يتم استخدام هذا المتغير بداخلها فقط

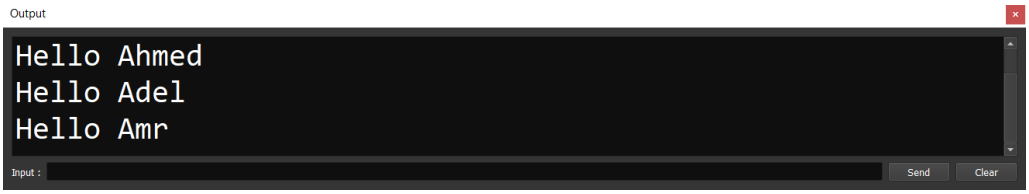
لاحظ! أن الدوال مفيدة لأنك تستخدمها أكثر من مرة وباستخدامات مختلفة لنفس الدالة

لاحظ! أن الدوال توفر الوقت والجهد، ومفيدة جداً ويجب التركيز المكثف عليها

الكود

```
say_hello( 'Ahmed' )
say_hello( 'Adel' )
say_hello( 'Amr' )

func say_hello( name )
    see( NL +  'Hello ' + name )
```

التنفيذ

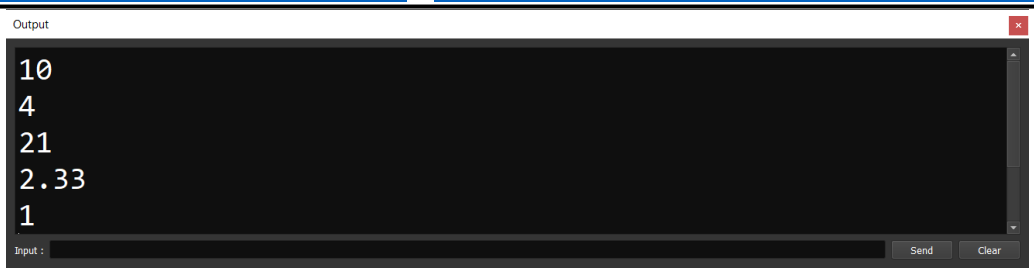


لاحظ! أنه فيما يلي تم انشاء دالة تستقبل الرقم الأول والرقم الثاني والعملية operation لتقوم بحساب ما تريد عند النداء عليها سواء كان جمع او طرح او قسمة او ضرب في نفس الدالة

الكود

```
calc( 7, 3, '+' )
calc( 7, 3, '-' )
calc( 7, 3, '*' )
calc( 7, 3, '/' )
calc( 7, 3, '%' )

func calc( num1 , num2 , ope )
    if ope = '+'
        see num1 + num2 + NL
    but ope = '-'
        see num1 - num2 + NL
    but ope = '*'
        see num1 * num2 + NL
    but ope = '/'
        if num2=0 num2=1 ok
        see num1 / num2 + NL
    but ope = '%'
        if num2=0 num2=1 ok
        see num1 % num2 + NL
    ok
```

التنفيذ



لاحظ! أنه فيما يلي تم ارسال ارقام للدالة لتطبع ناتج جمعهم

الكود

```
PrintSum( [1,2,3,4,5,6] )
PrintSum( [7,8,9,10,11] )
PrintSum( [12,13,14,15] )

func PrintSum( NumberList )
   mySum = 0
   for num in NumberList
      if lower( type(num) )='number'
         mySum += num
      ok
   next
   see mySum + NL
```

التنفيذ

انشاء دالة ترجع قيمة

لاحظ! أنه فيما يلي تم ارجاع قيمة، أي أنه عند النداء على هذه الدالة سوف ترجع قيمة يمكن طباعتها او وضعها في متغير

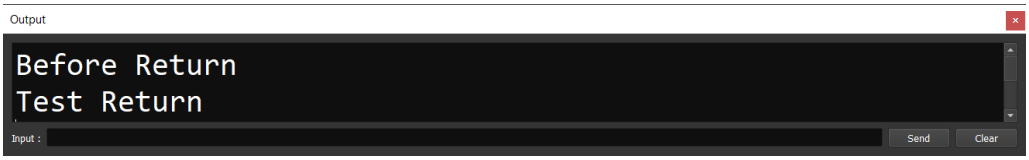
لاحظ! أن الكود بعد الارجاع return ليس له أي قيمة

لاحظ! أنه فيما يلي تم التعامل مع بعض الدوال التي تم عملها من قبل ولكن التغيير فيها أنها ترجع قيمة بدلا من أن تقوم بالطباعة، بحيث تأخذ انت القيمة وتفعل بها ما تريد

الكود

```
my_return_val = my_func()
see my_return_val + NL

func my_func()
    see 'Before Return' + NL
    return 'Test Return'
    see 'After Return' + NL
```

التنفيذ



الكود

```
see  say_hello('Ahmed') + NL
see  say_hello('Adel') + NL
see  say_hello('Amr') + NL

func say_hello( name )
    return 'Hello ' + name
```

التنفيذ

الكود

```
see PrintSum( [1,2,3,4,5,6] ) + NL
see PrintSum( [7,8,9,10,11] ) + NL
see PrintSum( [12,13,14,15] ) + NL

func PrintSum( NumberList )
   mySum = 0
   for num in NumberList
      if lower( type(num) )='number'
         mySum += num
      ok
   next
   return mySum
```

التنفيذ

ارسال الدالة كوسيط ثم تشغيلها

لاحظ! أنه فيما يلي تم بناء الدالة main التي تعمل تلقائياً ، وتم عمل دالة تستقبل وسيط دالة ثم تشغيلها وهذا النوع في لغة الرينج يسمي

First-class Functions، ولاحظ أن call للنداء على الدوال،
وأيضاً لاحظ أن الدالة يتم ارسالها بكولن : قبلها

الكود

```
Func Main
   see "before test2()" + nl
   f = Test2(:Test)
   see "after test2()" + nl
   call f()

Func Test
   see "Message from test!" + nl

Func Test2( f1 )
   call f1()
   See "Message from test2!" + nl
   return f1
```

التنفيذ

انشاء دالة تشغل الدوال بعدد من المرات

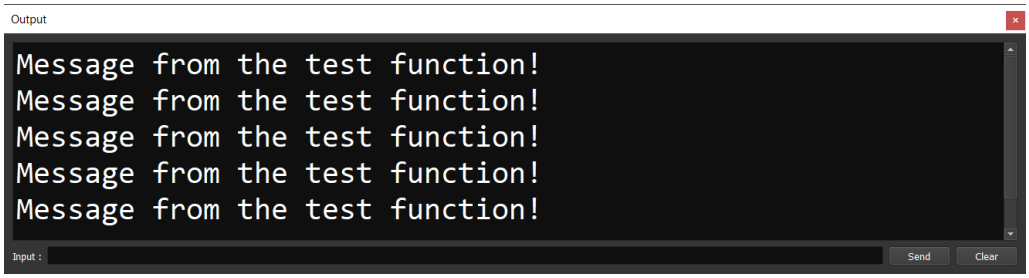
لاحظ! أنه فيما يلي تم عمل دالة باسم times تستقبل عدد ثم دالة،
بحيث تشغل الدالة بالعدد الذي تم تحديده، وهنا لاحظ ان الدالة test تم
عملها خمس مرات، ويمكنك تغيير العدد في أي وقت كما تحب مع أي دالة

بنفس التعامل السهل، وهذا النوع في لغة البرمجة رينج يسمي بالـ Higher-order Functions

الكود

```
Func Main
   times(5,:test)

Func Test
   see "Message from the test function!" + nl

Func Times( nCount , F )
   for x = 1 to nCount
      Call F()
   next
```

التنفيذ



استخدام الدوال المجهولة أو تعبيرات Lambda

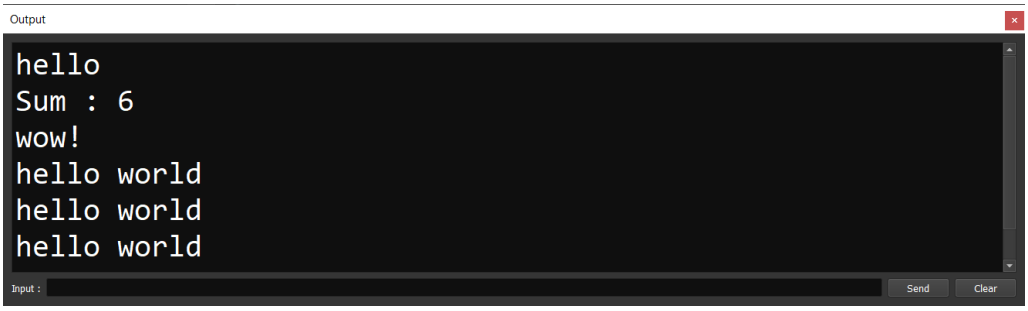
لاحظ! أنه فيما يلي تم وضع الدالة داخل الاقواس بدون ان تكون معرفة مسبقاً

لاحظ! أنه فيما يلي تم استخدام نمط مختلف للدوال وهو بدون أقواس، لأن رينج لغة الحرية، ويمكنك قراءة الدوال من ملف Help مع اللغة

الكود

```
test( func x,y {
      see "hello" + nl
      see "Sum : " + (x+y) + nl
      } )

times(3, func { see "hello world" + nl } )

func test x
   call x(3,3)
   see "wow!" + nl

func times n,x
   for t=1 to n
      call x()
   next
```

التنفيذ



انشاء دالة مرجعية تنادي على نفسها

لاحظ! أنه فيما يلي تم انشاء دالة لحساب وارجاع المضروب factorial

لاحظ! أن الدالة المرجعية لها فهم خاص، ولفهمها جيداً ننصح بالدخول على يوتيوب والبحث عن شرح أكاديمية حسونة للدالة المرجعية في أي لغة برمجة فلا تفرق لأن مفهومها واحد

الكود

```
see 'Factorial 3 is: ' + fact(3) + NL
see 'Factorial 4 is: ' + fact(4) + NL
see 'Factorial 5 is: ' + fact(5) + NL

func fact(num)
    if num = 0 return 1
    else return num * fact( num - 1 ) ok
```

التنفيذ

انشاء مكتبة أكواد قابلة للتنفيذ

أولاً انشاء ملف باسم myLib.ring في نفس مكان الملف التنفيذي الذي نجرب علية الأكواد الأساسية، أما الملف myLib.ring هو الذي يعبر عن المكتبة الجاهزة الاستخدام

ثانياً كتابة الكود التالي في الملف myLib.ring وحفظه

```
func createFolder(path)
   System(    'mkdir "' + path + '" '    )

func deleteFolder(path)
```

```
    System(     'rmdir "' + path + '" '     )

func getRandom(min, max)
    randNum = Random(  max - min  ) + min
    return randNum

func deleteAllFiles(path)
    all = dir(path)
    for x=1 to len(all)
        Remove(path + "/" + all[x][1]  )
    next

func deleteAllFolders(path)
    all = dir(path)
    for x=1 to len(all)
        if all[x][2]=1
            deleteAllFiles(path + "/" + all[x][1])
            deleteFolder(path + "/" + all[x][1])
        ok
    next

func avg(numbers)
    av = sum(numbers) / len(numbers)
    return av

func sum(numbers)
    if type(numbers)="LIST"
        s = 0
        for x=1 to len(numbers)
            if type(numbers[x])!="NUMBER"
                see "Error: Please Use Number Only" + nl
            else
                s += numbers[x]
            ok
        next
```

```
        return s
    else
        see "Please Send LIST Only" + nl
    ok

func isNumberList(numberList)
    for x=1 to len(numberList)
        if Type( numberList[x] ) != "NUMBER"
            return false
        ok
    next
    return true

func isStringList(stringList)
    for x=1 to len(stringList)
        if Type( stringList[x] ) != "STRING"
            return false
        ok
    next
    return true
```

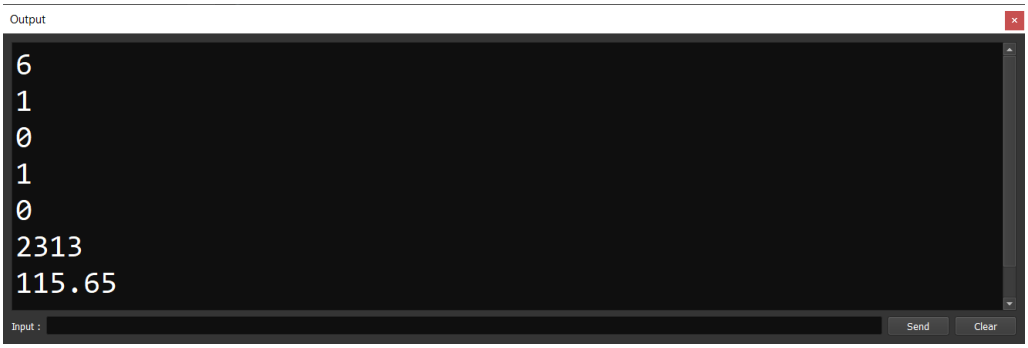
ثالثاً تجربة ما يلي على ملف الأكواد الأساسي

الكود

```
load "myLib.ring"

see getRandom(5,10) + NL

see IsNumberList( [1,2,3,4,5,6] ) + NL
see IsNumberList( [1,2,3,'Hello',4,5,6] ) + NL

see IsStringList( ['A','B','C'] ) + NL
see IsStringList( ['A','B',1999,'C'] ) + NL

nums = [55,66,987,45,22,336,654,88,9,6,5,4,7,1,2,3,6,5,4,8]
see sum(nums) + NL
```

```
see avg(nums) + NL
```

التنفيذ



تنفيذ أكواد من ملف خارجي أو من نص

لاحظ! أنه فيما يلي تم استخدام الدالة eval والتي تأخذ كود مكتوب في مقطع نصي لتقوم بتنفيذه وكأنك قمت بكتابته، ولتجربة ذلك اتبع ما يلي:

أولاً انشاء ملف باسم mycode.ring في نفس مكان الملف التنفيذي الذي نجرب عليه الأكواد الأساسية

ثانياً كتابة الكود التالي في الملف mycode.ring وحفظه

```
see 'Test From mycode.ring' + NL
for x=1 to 3
    see 'Test From mycode.ring ' + string(x) + NL
next

func calc( num1 , num2 , ope )
    if ope = '+'
        see num1 + num2 + NL
    but ope = '-'
        see num1 - num2 + NL
    but ope = '*'
        see num1 * num2 + NL
```

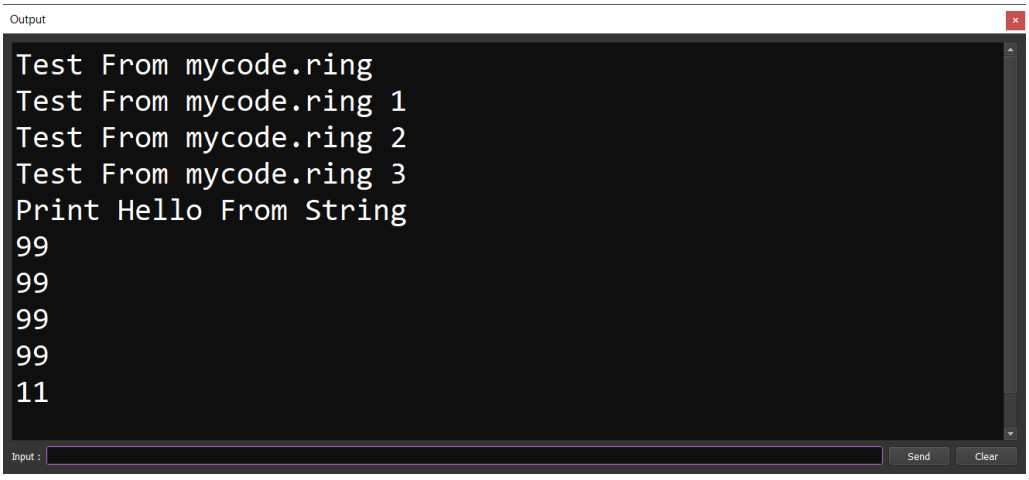
```
    but ope = '/'
        if num2=0 num2=1 ok
        see num1 / num2 + NL
    but ope = '%'
        if num2=0 num2=1 ok
        see num1 % num2 + NL
    ok
```

ثالثاً تجربة ما يلي على ملف الأكواد الأساسي

الكود

```
eval( read('mycode.ring') )
eval("see 'Print Hello From String' + NL ")
calc(81,18,'+')
calc(150,51,'-')
calc(33,3,'*')
calc(495,5,'/')
calc(23,12,'%')
```

التنفيذ



الكود

```
eval("
for x=1 to 3
```

```
    see 'X=' + string(x) + NL
next
")
```

التنفيذ

انشاء class به خصائص فقط

لاحظ! أنه فيما يلي تم عمل كلاس باسم employee ويحمل الرقم number والاسم name والعنوان address والمرتب salary والفاعلية active

لاحظ! أن الخصائص داخل الكلاس عبارة عن متغيرات

الكود

```
class employee
    number  = 0
    name    = ''
    address = ''
    salary  = 0.0
    active  = True
```

التنفيذ

لا يوجد شيء يتنفذ لأنه لم يتم استخدام الكلاس

انشاء class به وظائف فقط

لاحظ! أن الوظائف للكلاس هي الدوال التي تعاملنا معها من قبل

الكود

```
class my_print
    func print1 see 'Test Print 1' + NL

    func print2 see 'Test Print 2' + NL

    func print3 see 'Test Print 3' + NL
```

التنفيذ

لا يوجد شيء يتنفذ لأنه لم يتم استخدام الكلاس

انشاء class به خصائص ووظائف معاً

لاحظ! أنه فيما يلي تم النداء على عناصر الكلاس داخل الدوال بسهولة

الكود

```
class employee
    number  = 0
    name    = ''
    address = ''
    salary  = 0.0
    active  = True

    func get_data
        info =  string(number) + ';' + name + ';' + address
        info += ';' + string(salary) + ';' + string(active)
        return info
```

```
    func print_data
        see get_data() + NL
```

التنفيذ

لا يوجد شيء يتنفذ لأنه لم يتم استخدام الكلاس

انشاء class واستخدامه

لاحظ! أنه فيما يلي تم استخدام الكلاس عن طريق تعريف كائن object منه باسم emp1 عن طريق new أي نسخة جديدة من الكلاس

الكود

```
emp1 = new employee
emp1.number  = 1
emp1.name    = 'Adel'
emp1.address = 'Giza'
emp1.salary  = 9500.5
emp1.print_data()

class employee
    number  = 0
    name    = ''
    address = ''
    salary  = 0.0
    active  = True

    func get_data
        info =  string(number) + ';' + name + ';' + address
        info += ';' + string(salary) + ';' + string(active)
        return info
```

```
    func print_data
        see get_data() + NL
```

التنفيذ

انشاء class وعمل أكثر من object له

لاحظ! أنه تم ارسال متغيرين model وcolor للدالة set_data واستخدمنا this لنفرق بين ما بداخل الكلاس والتي تم ارسالها

الكود

```
car1 = new car
car2 = new car
car3 = new car

car1.set_data('Renault BMW','Red')
car2.set_data('Audi Mercedes Benz','Silver')
car3.set_data('MG Motor Maruti Suzuki','White')

car1.print_data()
car2.print_data()
car3.print_data()

class car
    model = ''
    color = ''
    func set_data( model, color)
        this.model = model
        this.color = color
    func get_data()
```

```
        return model+' , '+color
    func print_data()
        see  get_data() + NL
```

التنفيذ

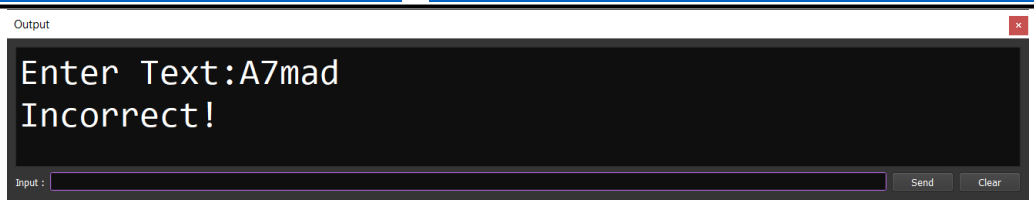
استخدام Regular Expression

لاحظ! أنه فيما يلي تم عمل نسخة جديدة بـ new من qRegularExpression لاستخدامها

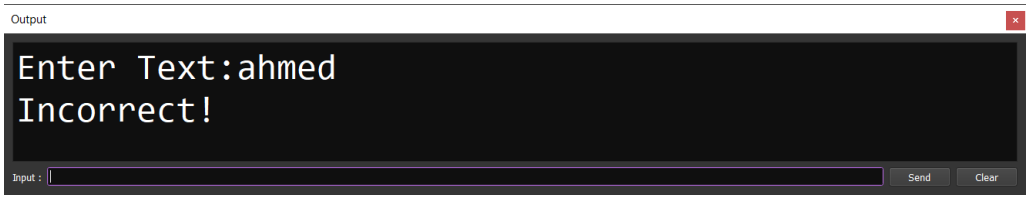
الكود

```
load "guilib.ring"

patt = '^[A-Z][a-z]{1,15}$'
see 'Enter Text:' give text

regex = new qRegularExpression()
regex.setPattern(patt)
match = regex.match( text ,0,0,0)

if match.hasMatch()
   see "Correct"
else
   see "Incorrect!"
ok
```

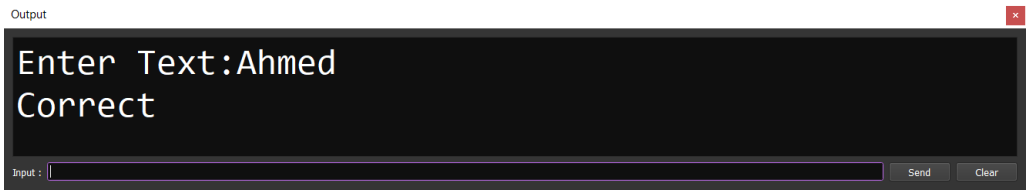
التنفيذ 1



التنفيذ 2



التنفيذ 3



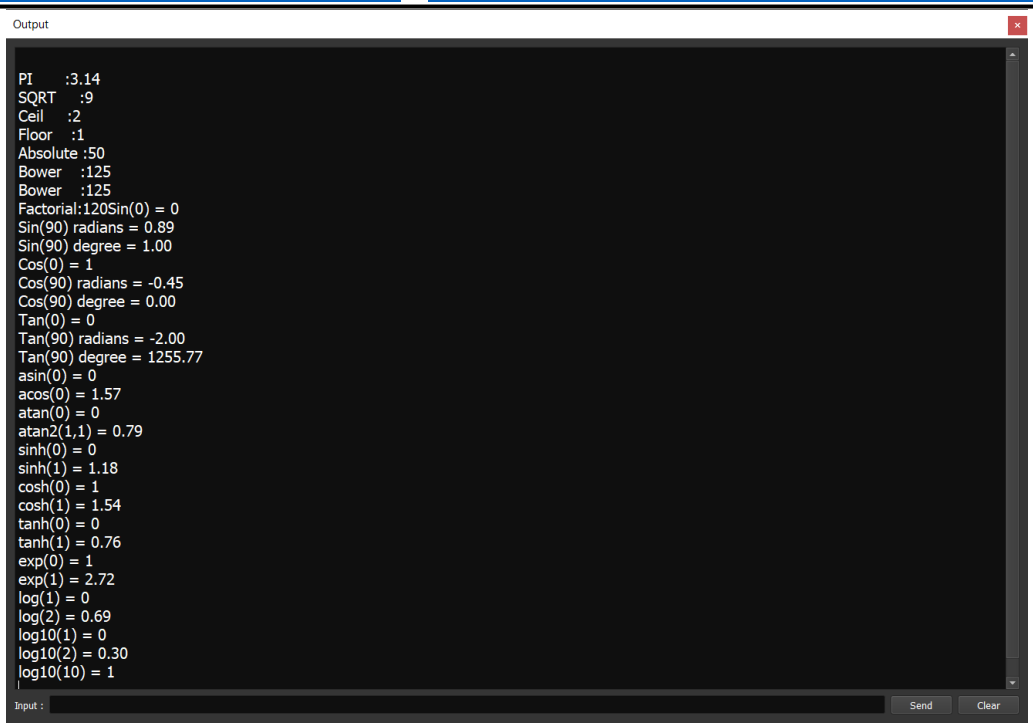
استخدام الرياضيات math

الكود

```
load "stdlib.ring"
oMath = new Math
see NL +  'PI       :' + 22/7
see NL +  'SQRT     :' + oMath.sqrt(81)
see NL +  'Ceil     :' + oMath.ceil(1.1)
see NL +  'Floor    :' + oMath.floor(1.99)
see NL +  'Absolute :' + oMath.fabs(-50)
see NL +  'Bower    :' + oMath.pow(5,3)
see NL +  'Bower    :' + pow(5,3)
see NL +  'Factorial:' + factorial(5)
See "Sin(0) = " + oMath.sin(0) + nl
See "Sin(90) radians = " + oMath.sin(90) + nl
```

```
See "Sin(90) degree = " + oMath.sin(90*3.14/180) + nl
See "Cos(0) = " + oMath.cos(0) + nl
See "Cos(90) radians = " + oMath.cos(90) + nl
See "Cos(90) degree = " +oMath. cos(90*3.14/180) + nl
See "Tan(0) = " + oMath.tan(0) + nl
See "Tan(90) radians = " + oMath.tan(90) + nl
See "Tan(90) degree = " + oMath.tan(90*3.14/180) + nl
See "asin(0) = " + oMath.asin(0) + nl
See "acos(0) = " + oMath.acos(0) + nl
See "atan(0) = " + oMath.atan(0) + nl
See "atan2(1,1) = " +oMath. atan2(1,1) + nl
See "sinh(0) = " + oMath.sinh(0) + nl
See "sinh(1) = " + oMath.sinh(1) + nl
See "cosh(0) = " + oMath.cosh(0) + nl
See "cosh(1) = " + oMath.cosh(1) + nl
See "tanh(0) = " + oMath.tanh(0) + nl
See "tanh(1) = " + oMath.tanh(1) + nl
See "exp(0) = " + oMath.exp(0) + nl
See "exp(1) = " + oMath.exp(1) + nl
See "log(1) = " + oMath.log(1) + nl
See "log(2) = " + oMath.log(2) + nl
See "log10(1) = " + oMath.log10(1) + nl
See "log10(2) = " + oMath.log10(2) + nl
See "log10(10) = " + oMath.log10(10) + nl
```

التنفيذ



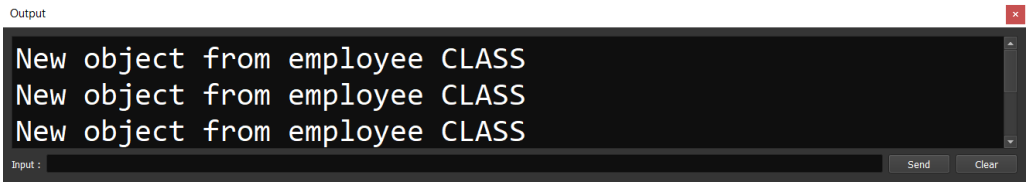
دالة البناء في الكلاس

لاحظ! أنه فيما يلي تم كتابة كود بدالة البناء init للكلاس employee وهذه الدالة يتم تنفيذها تلقائياً عند تعريف كائن جديد

الكود

```
emp1 = new employee()
emp2 = new employee()
emp3 = new employee()

class employee
    func init()
        see 'New object from employee CLASS' + NL
```

التنفيذ



وسائط مع دالة البناء

لاحظ! أنه فيما يلي تم ارسال الوسائط اجباري مع تعريف كل كائن من الكلاس لأنه يتم طلبها في دالة البناء، ولاحظ أن self تقوم بنفس دور this تماماً، وكأنك تنادي بهما على ما في داخل كل كائن من الكلاس

الكود

```
emp1 = new employee(1,'Ahmed')
emp2 = new employee(2,'Adel')
emp3 = new employee(3,'Amr')

emp1.print_data()
emp2.print_data()
emp3.print_data()

class employee
    emp_id   = 0
    emp_name = ''

    func init(emp_id, emp_name)
        self.emp_id   = emp_id
        self.emp_name = emp_name

    func print_data()
        see string(emp_id) + ' , ' + emp_name + NL
```

التنفيذ

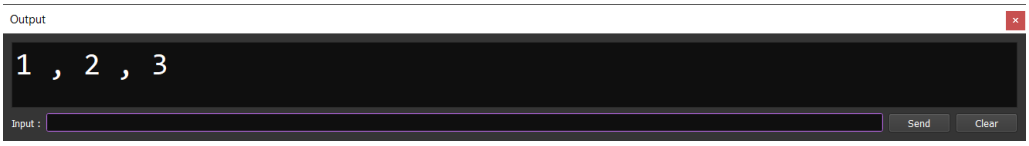
انشاء عداد وindex للـ objects

لاحظ! أنه فيما يلي تم عمل عداد لعدد الكائنات objects_count
وكل كائن يحتفظ بالـ index الخاص به

الكود

```
count = 0
e1 = new employee()
e2 = new employee()
e3 = new employee()
see string(e1.index) + ' , ' + e2.index + ' , ' + e3.index

class employee
    index
    func init()
        count += 1
        index = count
```

التنفيذ



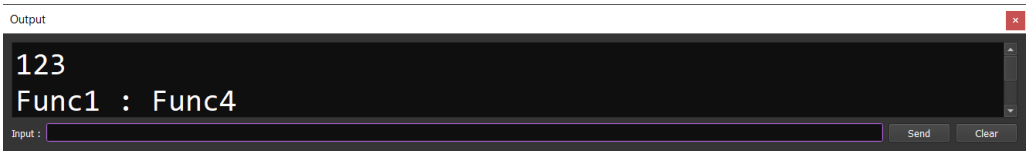
انشاء عناصر سرية داخل الـ class

لاحظ! أن كل ما تحت العامل private أصبح سري بداخل الكلاس فقط ولا يتم استدعاؤه من أي كائن ولو تم سيحدث خطأ لأنه سري

الكود

```
func main
   e = new emp
   e.att0 = "123"
   see e.att0 + nl
   e.func1()

class emp

   att0
   att00

   func func1 see "Func1 : " + func4() + nl
   func func2 see "Func2" + nl
   func func3 see "Func3" + nl

   private
   att1
   att2
   att3

   func func4 return "Func4" + nl
   func func5 see "Func5" + nl
```

التنفيذ



الوراثة

لاحظ! أنه تمت عملية وراثة بين person وبين employee حيث أن الـ employee ورث كل ما في الـ person وذلك بـ from وبعدها person بعد تعريف employee

الكود

```
emp1 = new employee
see( NL +  emp1.name )
see( NL +  emp1.address )
see( NL + '==============')
emp1.seedata()

class person
    name = 'Person'
    address = 'Egypt'

    func seedata()
        see( NL +  self.name + ' ; ' + self.address )

class employee from person
```

التنفيذ

اظهار اسم الكلاس

الكود

```
e = new employee
d = new doctor
s = new student
see classname(e) + NL
see classname(d) + NL
see classname(s) + NL

class employee
class doctor
class student
```

التنفيذ

إعادة الكتابة على الدوال

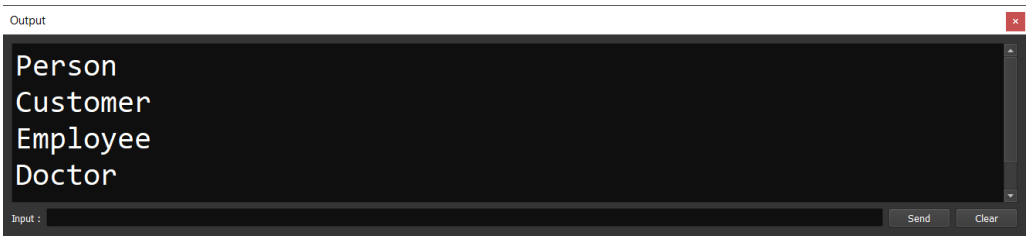
لاحظ! أنه فيما يلي تم إعادة الكتابة على الدوال عن طريق كتابة الدالة مرة اخري في الكلاس الذي ورث لتكون مخصصة له

الكود

```
p = new person
c = new customer
e = new employee
d = new doctor

p.printtype()
```

```
c.printtype()
e.printtype()
d.printtype()

class person
    func printtype()
        see 'Person' + NL

class customer from person
    func printtype()
        see 'Customer' + NL


class employee from person
    func printtype()
        see 'Employee' + NL

class doctor from employee
    func printtype()
        see 'Doctor' + NL
```

التنفيذ



الحزم Packages

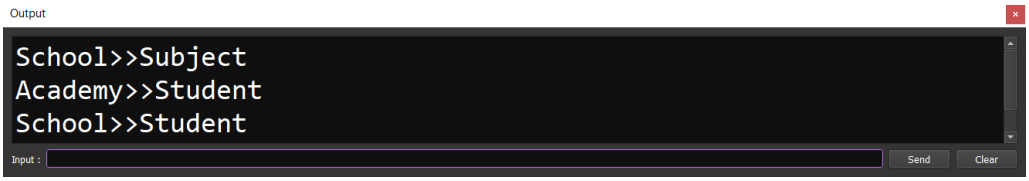
لاحظ! أنه فيما يلي تم حزم packages لتكون بمثابة حاوية، فكما تم وضع المتغيرات والجمل في الدالة، وتم وضع الدوال والمتغيرات في الكلاس، فسيتم وضع الكلاسات في حاوية له وهي الحزمة

لاحظ! أنه فيما يلي تم المناداة على اسم الحزمة ثم اسم الكلاس، ولاحظ أيضاً انه من الممكن عمل import للحزمة للنداء على الكلاس مباشرة

الكود

```
import academy
import school

func main
sub1 = new subject
sub1.getType()

std1 = new student
std1.getType()

std2 = new school.student
std2.getType()

package school
class subject func getType see "School>>Subject" + nl
class student func getType see "School>>Student" + nl
class doctor func getType see "School>>Doctor" + nl

package academy
class student func getType see "Academy>>Student" + nl
class doctor func getType see "Academy>>Doctor" + nl
```

التنفيذ



موضوع Ring Package Manager مفيد جداً

وبعض البرامج التي تم عملها بـ Ring

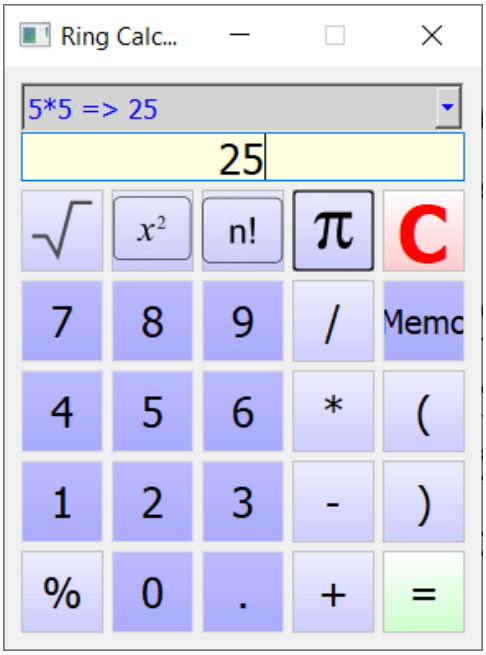
وبعض الألعاب Games التي تم عملها بـ Ring

يوجد ضمن اللغة الـ Ring Package Manager أو RingPM وهو نظام في اللغة يظهر لك التطبيقات والألعاب التي تم عملها بلغة الرينج وهو محدث دائماً بفضل الله، ولكن تأكد أنك قمت بتنزيل آخر اصدار من لغة الـ Ring ولرؤيتها بسهولة كما يلي:

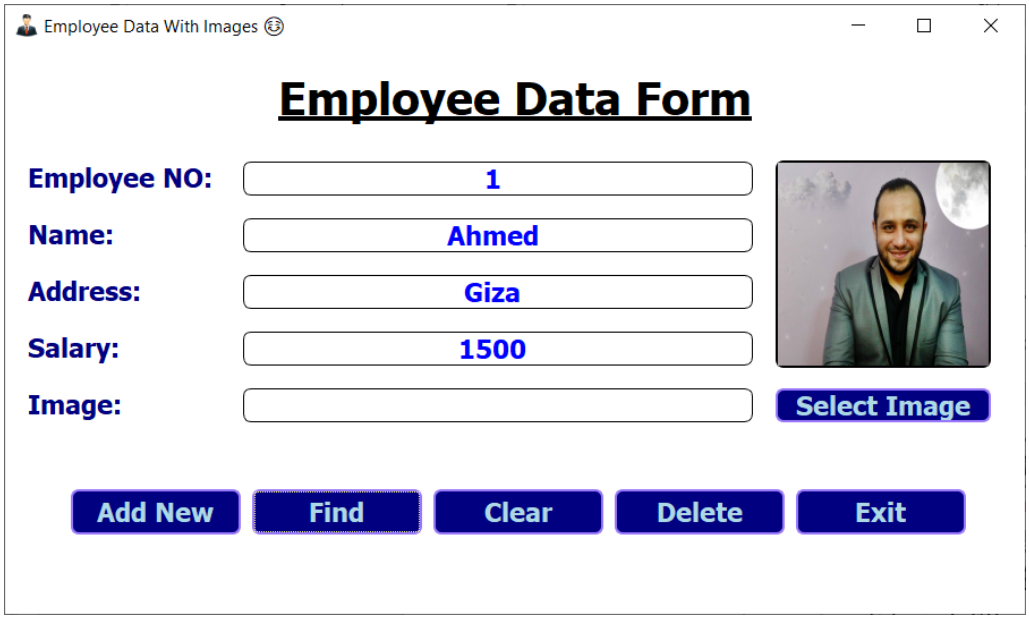
1. فتح منفذ الأوامر سواء CMD أو Terminal
2. كتابة الأمر ringpm list

وستجد كل المحتويات بأسمائها، مع اسم كل عنصر بين قوسين ()، حيث أن كل هذه العناصر موجودة في مجلد اللغة باسم ring ثم مجلد ringpm ثم مجلد packages، ويمكنك فتح أي واحده منهم لتري أكوادها وتستفاد من خبرات عالمية تعاونية من فريق اللغة، حيث أنك لو أخذت الاسم وكتبت ringpm ثم مسافة ثم run ثم مسافة ثم اسم العنصر سيتم تشغيله فوراً .

على سبيل المثال نكتب ringpm run calculator وستعمل الآلة



أو نكتب ringpm run employee وسيعمل برنامج الموظف الذي تم شرح عمله بالكامل في اكاديمية حسونة



ويمكن تشغيل اللعبة GoldMagic800 التي تم شرح اللعب معها في حلقة رقم 144 من كورس اللغة Ring على اكاديمية حسونة، حيث نكتب الأمر ringpm run goldmagic800 وستعمل

كما يمكنك أن تكتب الأمر في الرينج بشكل مباشر عن طريق system كما تعلمنا من قبل مثل الكود التالي:

```
system('ringpm run goldmagic800')
```

والتنفيذ سوف يشغل اللعبة وكأنك فتحتها على منفذ الأوامر

كنوز وهدايا

اليك كنوز وهدايا، ووصيتي لك، أن تنشرها في كل مكان من الروابط الأصلية، وأيضاً وصيتي لك بأن تشاهدها وتنصح الجميع بمشاهدتها من اليوتيوب من أكاديمية حسونة، لتنجح هذه الكورسات بفضل الله 😊.

شرح في قصص نجاح

https://www.youtube.com/playlist?list=PLHIfW1KZRIflvUNrnmYsLPbU2t4oG0P-8

شرح في الكمبيوتر بشكل عام

https://www.youtube.com/playlist?list=PLHIfW1KZRIfm_dFhGnawAlnOOE1aSwoSN

شرح عن مقدمة في الحاسبات

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmt3FFalD6y-UjTshUC_zXM

شرح في أنظمة ماك ولينكس

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmTj0gddZTgcPdqDGP1qscF

شرح في الويندوز Windows من الأساس وبالتفصيل

https://www.youtube.com/playlist?list=PLHIfW1KZRIfkKu8K8Dz603sxLa3eQ-g6G

شرح في برنامج وورد Word من الأساس

https://www.youtube.com/playlist?list=PLHIfW1KZRIfk_g1_xWEJ8gl_nbPY_Y75A

شرح في برنامج اكسيل Excel من الأساس

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmeaZUramzmRwJVLQThBiuu

شرح في برنامج الفوتوشوب Photoshop
https://www.youtube.com/playlist?list=PLHIfW1KZRIfntLWyOJQKyfR5RcuXdzfnU

شرح في البرامج الجاهزة
https://www.youtube.com/playlist?list=PLHIfW1KZRIfmCe0esfmzTZTQ_AfoLOH1M

شرح في الخوارزميات وخرائط التدفق + أمثلة
https://www.youtube.com/playlist?list=PLHIfW1KZRIfmOvX3-91zteoSP-b0VS44o

شرح آخر في الخوارزميات وخرائط التدفق
https://www.youtube.com/playlist?list=PLHIfW1KZRIflwIO8IiPxNkSSb-ezkNqDs

شرح في تحليل البيانات
https://www.youtube.com/playlist?list=PLHIfW1KZRIfkDoNSoObO0s_OZn67DFAti

برنامج المبرمجتية
https://www.youtube.com/playlist?list=PLHIfW1KZRIfnLcjFtaI9O5vt1DKdSVM4K

برنامج أساس مبرمج
https://www.youtube.com/playlist?list=PLHIfW1KZRIfkb-fDWH14OjJowQX_Rztgk

شرح في لا اعرف شيء عن الكمبيوتر وأريد أن اكون مبرمج
https://www.youtube.com/playlist?list=PLHIfW1KZRIfnKiRVeW9H2k4TnwXZbaoc8

شرح في منهج الصف الثاني الاعدادي

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmkYyJx1FJcSx1QOJnHanpQ

شرح في منهج الصف الثالث الاعدادي

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmM2mbNfA_4jTI3vyK4iCOE

شرح في كورس قديم عن الويب HTML

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmpHW37aNxMFosXD_p1jQz0

شرح في كورس جديد عن الويب HTML

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnXKjZ9UcT5BVt9x19BlItX

شرح في كورس عن الويب CSS

https://www.youtube.com/playlist?list=PLHIfW1KZRIflO6xnz4jpmzzSuo5B63aJf

شرح في كورس عن الويب JavaScript

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmiYYwt29BG7N9zzsd-jLRQ

شرح في دردشة برمجة عملي

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl0CJzVbN-DHGUJ_Y2DpPim

شرح في الويب جي كويري j-query

https://www.youtube.com/playlist?list=PLHIfW1KZRIfll3ObMFi02Ry7oRU3MJVRG

شرح في الوي بوت ستراب bootstrap

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmzBcYnuLNrmdLxYCDzJgQi

شرح في جاسون JSON

https://www.youtube.com/playlist?list=PLHIfW1KZRIflA5jVQbAAHVizSxoeOAtgq

شرح في انجولار Angular

https://www.youtube.com/playlist?list=PLHIfW1KZRIflbtKBrCp5U2UdbeV1c7F-5

شرح في الويب asp.net وسي شارب ومع الخدمات الأساسية

https://www.youtube.com/playlist?list=PLHIfW1KZRIflkDPHlOLTKV1daEp4pvIgA

شرح في الويب ومع AJAX

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmYd8YFZtSmCzc3hcZXVua9

شرح في قواعد البيانات MySQL

https://www.youtube.com/playlist?list=PLHIfW1KZRIflO4GPp9vMtUgry-xAQKl7i

شرح في الويب ولغة البرمجة PHP

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmPDQahhTDdd1DJ4PwQl3pC

شرح في الويب ومع دبلومة الويب الكاملة

https://www.youtube.com/playlist?list=PLHIfW1KZRIflYXyXHNiG-hyyVaz7-9ujn

شرح في الويب ومع سلسلة تعلم الفيديو الواحد

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnPUohWz2vPjm0obEjzZJD8

شرح في الويب ومع صناعة آلة حاسبة بسيطة

https://www.youtube.com/playlist?list=PLHIfW1KZRIfkYSpySS9Czt3Dxx2rsvqcN

شرح في البرمجة واساسيات السي شارب C#

https://www.youtube.com/playlist?list=PLHIfW1KZRIfm8nQAoJF5u2aV43tMRAAmr

شرح في شاشات السي شارب C#

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnbNoGB0NdoRdllq9fdo6uM

شرح في البرمجة كائنية التوجه OOP وسي شارب C#

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl6UP-PlUli03pokSc4af2S

شرح في الويب مع سي شارب وC# ASP.NET

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmi8jwSbqiQuVfxeXIBk-bW

شرح في عالم واسع من قواعد البيانات DB مع C#

https://www.youtube.com/playlist?list=PLHIfW1KZRIflAus00vgdVEzLUBCx8ooH_

شرح في أمثلة عملية مع قواعد البيانات و#C

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmYdNXeeaBjt4mxSX5RWjzo

شرح في أسرار قواعد البيانات Database

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnKHFH1uUwdpeW8h-A8lAI_

شرح في الـ LINQ مع C#

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnW4RDln5tzw6htvNhnkr7t

شرح في عمل برنامج المطاعم والأكلات
https://www.youtube.com/playlist?list=PLHIfW1KZRIfnK119pOw2oEDXc0AEpIfd5

شرح في عمل برنامج المبيعات الشامل
https://www.youtube.com/playlist?list=PLHIfW1KZRIfm2vgnbTfoxoUSlxoag0UHX

شرح في دورة السي شارب العملاقة بفضل الله
https://www.youtube.com/playlist?list=PLHIfW1KZRIfkDF2xTIB5kX8gdthmLTufx

شرح في قواعد البيانات الشامل والسي شارب C#
https://www.youtube.com/playlist?list=PLHIfW1KZRIfmg0C_6ON0IRFWrifni6CTq

شرح في ربط MySQL مع C#
https://www.youtube.com/playlist?list=PLHIfW1KZRIfkQ7fgtqD4b4VuhxV6RzNu-

شرح في ربط Oracle مع C#
https://www.youtube.com/playlist?list=PLHIfW1KZRIfl0vnBDg28R0DW9HBkyHvc-

شرح في عمل متصفح بالسي شارب
https://www.youtube.com/playlist?list=PLHIfW1KZRIfl7ToU9EDuRVYiki9C9wxnJ

شرح في كورس سي شارب مختصر
https://www.youtube.com/playlist?list=PLHIfW1KZRIfmIhzWCyXnYlxz8ALlfPRAn

شرح في الويب ومع الدورة التنفيذية في سي شارب وASP.NET

https://www.youtube.com/playlist?list=PLHIfW1KZRIfn6kI3NyvzTDm-wdw9EX7D6

شرح في البرمجة ومع اساسيات فيجوال بيسك VB.NET

https://www.youtube.com/playlist?list=PLHIfW1KZRIflKvIMpDSgFTJsR_IO1dH9i

شرح في شاشات الفيجوال بيسك VB.NET

https://www.youtube.com/playlist?list=PLHIfW1KZRIfn7Pz7MnzABWryuHRP9ab7O

شرح في دورة الفيجوال بيسك العملاقة بإذن الله

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmLGhFsQ1G5cPv1xokzBsbT

شرح في فيجوال بيسك وسكول سيرفر مع التقارير الكاملة

https://www.youtube.com/playlist?list=PLHIfW1KZRIfkkhf3ha77qccAWYMvgM7LR

شرح في ربط VB.NET مع Oracle

https://www.youtube.com/playlist?list=PLHIfW1KZRIflAkiMFHY-0Jc_XXnpLWQO0

شرح في ربط VB.NET مع Access

https://www.youtube.com/playlist?list=PLHIfW1KZRIfkdbHPdTBpcYZzPQtPUeR7I

شرح في الفيجوال بيسك القديم VB 6

https://www.youtube.com/playlist?list=PLHIfW1KZRIfk5L0DQbV8DyUwfoAhQyieg

شرح في البرمجة مع أساسيات لغة الجافا JAVA

https://www.youtube.com/playlist?list=PLHIfW1KZRIflflvzYY7B1d5CJjXpbIriU

شرح في بداية البرمجة الكائنية مع جافا JAVA

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmb25mVAsCJy_Ehgu5rbbQx

شرح في البرمجة الكائنية مع جافا JAVA

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnT2i7Ba4F2nrWb_ENXjyTs

شرح في جافا مع قواعد البيانات

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnKQjOoZc540z3fmGe-slLU

شرح في جافا مع اندرويد Android

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnQ6WRvLbWok4ZZww4k3T1u

شرح في شاشات الجافا

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl2XxZwryDuPO5ZLkRhL-CS

شرح في دورة الجافا العملاقة بفضل الله

https://www.youtube.com/playlist?list=PLHIfW1KZRIfn9BnepQuzWiM_ZPIwUDawL

شرح في أمثلة عن جافا

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnUMvey6UAveK5msn8L_ulS

شرح في التعامل مع ملف الـ jar للجافا

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnDa6squg4OjDjYfx5YC8Zx

شرح في البرمجة مع لغة البرمجة كوتلن Kotlin

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl5UHtM6DRFAVn3PBuJ7NGD

شرح في كلام عن لغة كوتلن Kotlin

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl_KgeClVL_xH_JXFqIsV0Y

شرح في كوتلن مع اندرويد Android

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmbpu6cVCxo98u_Yk0Hfxah

شرح في البرمجة ومع اساسيات الرينج Ring

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmm35tXSbxsFDouRJAPP9Y2

شرح في البرمجة الكائنية OOP مع الرينج

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmHvfmFZZ0XuzZYuoosG9kl

شرح في البرمجة مع لغة الرينج Ring اساسيات وOOP

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnVNMUcSLFFyDzuLDheF99n

شرح في شاشات الرينج Ring

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmMhy8GltKHW5UJ_ZPnGJ0K

شرح في دورة الرينج العملاقة بفضل الله

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnM9y0sQRwjVz2-IwvnEJep

شرح في البرمجة مع لغة الرينج Ring

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl6KzfLziFl650MmThnQ0jT

شرح في البرمجة الكائنية OOP ورينج Ring

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl5TA4nTIIVWsfDyVwIBP6y

شرح في شاشات رينج Ring

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmE8gCtJQpIOGmBz3bc_Nrs

شرح في البرمجة مع لغة السي بلس بلس C++

https://www.youtube.com/playlist?list=PLHIfW1KZRIfkz_N9aNWXRByEJOudCxSRh

شرح متنوع وعام في عالم البرمجة

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl6aI0bGb0zWZyGHc2ctcW7

شرح في قواعد البيانات SQL Server

https://www.youtube.com/playlist?list=PLHIfW1KZRIflvUe-YfDpIqqnU70BopQ3q

شرح في قواعد البيانات ولغة السكول SQL

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmNd2URuqVO5Itboj_LQyrx

شرح في قواعد البيانات مع أكسس Access

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnKpy1msMUPqVYdpu3nOdkY

شرح في قواعد بيانات الوسائط المتعددة Multimedia DB

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmMYvE8mnEH4VT-4v21XV6p

شرح في عمل برنامجك مع الأداة Magic DB

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnkb0hIry8E0LbA0flhaXTo

شرح في برنامج امتحان الطلاب

https://www.youtube.com/playlist?list=PLHIfW1KZRIfnd5v0u8hxwTKXebJK9QpqG

تسالي مبرمجين

https://www.youtube.com/playlist?list=PLHIfW1KZRIfl_zc__duOyEhaCCPZRAuHv

اكاديمية حسونة

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmxKFdzlyAna7sYE_L2hI0o

لقاءات وأحداث Events

https://www.youtube.com/playlist?list=PLHIfW1KZRIfmFsmWKCbc_rf6qAPbVykMJ

شكر خاص

الشكر أولاً وأخيراً وابتداء وانتهاء لله عز وجل، ونشكر كل من:

- شكر لأبي وأمي اللذان ربياني صغيرا.
- شكر لكل أصدقائي واصحابي واحبابي في كل مكان.
- شكر لكل من عرف قدر العلم.
- شكر لكل طالب او طالبة او اخ او صاحب او صديق او حبيب ينشر هذا العمل في كل مكان ليفيد الجميع بإذن الله.
- شكر خاص للأستاذ احمد محمد باريان على دعمه في نشر هذا الكتاب ونشر العلم في كل مكان.
- شكر للمهندس أحمد إبراهيم سالم (Ahmed Ibrahim)
- شكر للدكتور عادل عبد الصبور (Adel Sabour).
- شكر للمهندس محمود سمير فايد (Mahmoud Fayed).

نشكرك على قراءتك

أولاً: نتقدم لك بخالص الشكر 💖 على قراءتك هذا الكتاب، ونتمنى لك الاستفادة الكاملة من محتواه.

ثانياً: لا تنسى أن تسجل لنفسك حساب 💪 في أكاديمية حسونة لتنضم إلينا وتشارك معنا كل محتويات الموقع، سجل من الرابط التالي:
https://www.hassouna-academy.com/register

ثالثاً: يسر أكاديمية حسونة أن تخبرها برأيك عن هذا الكتاب من خلال تقديم كلمة شكر 😍 من الرابط التالي:
https://www.hassouna-academy.com/thanks

References المراجع

الجدول التالي يحتوي على المراجع التي تم الرجوع إليها أثناء تأليف هذا الكتاب، سواء كانت مراجع تعليمية مرئية أو مقروءة، أو معلوماتية.

المرجع	الرابط
اكاديمية حسونة	https://www.hassouna-academy.com/
مخترع لغة الرينج م/محمود سمير فايد	http://www.ring-lang.net

The End الختام

والآن، أتوجه الى أحبابي بكلمة من القلب 💖، خالصة إن شاء الله، فكر معي في اختراعات الإنسان، فكر في تطويرات وصناعات الإنسان، فكر في الإنسان وقدراته، وقل سبحان الله الذي خلق الإنسان، فكر في قول الله عن الإنسان في القرآن الكريم "لَقَدْ خَلَقْنَا الإِنْسَانَ فِي أَحْسَنِ تَقْوِيمٍ"، فكر في قدرة الله الواحد الأحد، الفرد الصمد، الذي لم يلد ولم يولد، ولم يكن له كفؤاً أحد، فكر في الله العظيم، الذي خلق السماوات والأرض، فكر في رب العزة، فكر في رب الكون، وقل، سبحان الله، وقل، سبحان الملك العظيم، وقل سبحان ذي العزة والجبروت، سبحان ذو الملك والملكوت، سبحان الله العظيم، سبحان الله الكريم، سبحان الله، عدد خلقه، ورضا نفسه، وزنة عرشه، ومداد كلماته، وآخر دعوانا "الحمد لله رب العالمين".