الأكاديمية العربية الدولية
Arab International Academy

إسم المادة: مدخل إلى لغة البرمجة - بايثون

إسم المدرب : المهندس عبدالقادر عبدالله

الأكاديمية العربية الدولية – منصة أعد

الأكاديمية العربية الدولية
Arab International Academy

محتويات المحاضرة :

- تعريف عام بالبرمجة
- تعريف عام بلغة بايثون
- مجالات استخدام لغة بايثون
- اهم المحررات للتعامل مع لغة بايثون
- امر الطباعة في اللغة واهميته
- المتغيرات وانواعها
- العمليات الحسابية
- الجملة الشرطية والاحوال المتعددة
- الحلقات المتكررة
- المصفوفات
- الدوال وانواعها
- التعامل مع الفئات
- مدخل إلى البرمجة كائنية التوجه

تعريف عام بالبرمجة

قبل تعريف البرمجة بشكل مباشر لابد من الإشارة إلى وظيفة الحاسوب الاساسية , والمتصلة بشكل مباشر بتسميته , وذلك أنه يمتلك سرعة عالية جدا بالقيام بالعمليات الحسابية مقارنةً بالإنسان .

للتمكن من استخدام هذه السرعة العالية لابد من توجيه امر او عدة اوامر للحاسوب من شأنها أن تخبره بما يجب عليه أن يفعل وما النتيجة التي يجب الوصول إليها

البرمجة هي عملية ترتيب وتوجيه هذه الاوامر للحاسوب بشكل منظم بهدف بناء برنامج يقوم بأداء مهمة واحدة او عدة مهام .

يهدف الإنسان من خلال البرمجة إلى توظيف سرعة الحاسوب العالية لتنفيذ خوارزميات معينة .

تتم عملية البرمجة باستخدام إحدى لغات البرمجة والتي يختارها المبرمج مسبقا لتنفيذ الخوارزميات وبناء البرامج

تعرف لغات البرمجة على انها لغات التواصل بين الإنسان والآلة وتنقسم من حيث التصنيف إلى لغات عالية المستوى واخرى ضعيفة المستوى

تعريف عام بلغة بايثون

بايثون هي لغة برمجة متعددة الاستخدامات وذات تصميم بسيط وسهل القراءة، مما يجعلها مناسبة للمبتدئين والمحترفين.
طُوّرت في أواخر الثمانينيات على يد "غيدو فان روسوم" وتهدف إلى تبسيط عملية البرمجة مع الحفاظ على قوة الأداء.

تعتبر بايثون لغة ديناميكية، حيث يمكن استخدامها لبناء تطبيقات متنوعة تشمل تطبيقات سطح المكتب، مواقع الويب، وحتى برامج الذكاء الاصطناعي.

بالإضافة إلى ذلك، تمتاز بوجود مكتبات ضخمة تجعل تطوير المشاريع أسرع وأكثر كفاءة.

مجالات استخدام لغة بايثون

تُستخدم بايثون في مجالات متعددة نظرًا لمرونتها ودعمها الواسع.

من أشهر استخداماتها تطوير الويب باستخدام أطر مثل Django وFlask، وتحليل البيانات باستخدام مكتبات مثل Pandas وNumPy، والتعلم الآلي والذكاء الاصطناعي مع مكتبات مثل TensorFlow و Scikit-learn.

كما تُستخدم في الأتمتة، واختبار البرمجيات، وبناء أدوات صغيرة تساعد في تحسين سير العمل.

هذه الاستخدامات تجعل بايثون لغة مفضلة للعديد من الصناعات البرمجية

اهم المحررات للتعامل مع لغة بايثون

هناك العديد من المحررات التي تدعم بايثون بشكل ممتاز، وتختلف حسب احتياجات المطور.

باي - شارم : هو محرر متكامل يحتوي على أدوات تحليل وتصحيح متقدمة

في اس كود : يوفر بيئة خفيفة وسهلة الاستخدام مع دعم إضافات قوية

جوبيتر نوت بوك : يفضله علماء البيانات بسبب سهولة العمل مع النصوص البرمجية وتنفيذ الخلايا التفاعلية

سب لايم تيكست : يعد خفيفًا وسريعًا، ويحتوي على إضافات تدعم بايثون هذه المحررات تساعد على تسهيل كتابة وفحص الكود بشكل سريع وفعال

امر الطباعة في اللغة واهميته

يعد أمر الطباعة في بايثون من أهم الأوامر التي تستخدم لعرض المعلومات على الشاشة.

يوفر للمبرمج وسيلة لفهم كيفية عمل البرنامج واختبار النتائج.

سواء كان الأمر يعرض نتائج العمليات الحسابية أو الرسائل النصية للمستخدم، فإنه يساعد على تصحيح الأخطاء وتتبع سير البرنامج في بيئة التطوير.

يمكن أيضًا استخدام الطباعة لمراقبة القيم المتغيرة أثناء تنفيذ البرنامج، مما يسهم في تحسين الأداء العام واكتشاف المشكلات

يكتب امر الطباعة بشكل بسيط كالتالي :

```
print()
```

المتغيرات وانواعها

المتغيرات في بايثون هي حاويات لتخزين البيانات التي يمكن استخدامها لاحقًا في البرنامج.

يمكن أن تكون المتغيرات من انواع مختلفة مثل الاعداد الصحيحة والعشرية والنصوص والقيم المنطقية.

يتم إنشاء المتغيرات ديناميكيًا في بايثون، حيث لا يحتاج المبرمج لتحديد نوع المتغير مسبقًا.

تعطي المرونة في التعامل مع المتغيرات المبرمجين القدرة على بناء برامج معقدة بسهولة أكبر، وتحسين قابلية القراءة.

العمليات الحسابية

مثل العديد من اللغات البرمجية تتمتع لغة بايثونن بسهولة القيام بالعمليات الحسابية كونها قريبة من لغة البشر (عالية المستوى).

وتجب الإشارة إلى أن العمليات الحسابية تخضع للقوانين الرياضية العامة
حيث أن الاولوية للاقواس ثم للضرب والقسمة واخيرا للجمع والطرح

في البرمجة يمكن إسناد قيمة للمتغير بناءً على قيمته القديمة مما يعني أنه من الممكن أن تجد امر برمجي بالصورة التالية :

```
X = X + 1 ;
```

حيث تصبح هنا قيمة X الجديدة تساوي قيمتها القديمة + 1

توفر بايثون أيضًا دعمًا للعمليات المنطقية والمقارنات.

يمكن استخدام هذه العمليات على أنواع البيانات العددية أو النصية، مما يتيح للمطورين تنفيذ حسابات معقدة بسهولة

الجملة الشرطية

تعد الجملة الشرطية واحدة من اهم المفاهيم البرمجية بشكل عام والتي يمكن تنفيذها باستخدام لغة جافا بسهولة

يعتمد الشرط بشكل اساسي على قيمة منطقية (حقيقي – غير حقيقي) ويتم الحصول على هذه القيمة اما بشكل مباشر (متغير من نوع boolean)

او عبر اجراء مقارنة بين متغيرين او اكثر

```
3  x = 10
4  if x == 10 :
5      print('X value is 10')
```

ممكن اضافة عدة شروط متصلة ببعضها باستخدام عبارات شرطية
مثل
elif
else
للتحقق من اكثر من حالة في ذات الوقت

الحلقات المتكررة

الحلقات المتكررة في لغة بايثون تُستخدم لتنفيذ مجموعة من التعليمات بشكل متكرر طالما أن شرطًا معينًا محقق.
هناك نوعين من الحلقات:
حلقة (for) : تُستخدم عندما تعرف عدد التكرارات مسبقًا، وتتيح التحكم في المتغيرات المستخدمة في التكرار.

حلقة (while) : تستمر في التكرار طالما أن الشرط المحدد صحيح، وتستخدم غالبًا عندما لا يكون عدد التكرارات معروفًا.

تسهل الحلقات تنفيذ العمليات المتكررة وتوفير الوقت والجهد مقارنة بكتابة نفس التعليمات عدة مرات.

تستخدم الحلقات بشكل واسع في معالجة البيانات، حيث تمكن المطورين من معالجة كل عنصر في قائمة أو مصفوفة

المصفوفات

المصفوفات في لغة بايثون هي هيكل بيانات يُستخدم لتخزين مجموعة من العناصر المتشابهة في النوع (مثل أرقام أو نصوص) في مكان واحد.

المصفوفات (أو القوائم في بايثون) هي نوع من البيانات التي تحتوي على مجموعة من العناصر المرتبة.

يمكن أن تكون هذه العناصر من أي نوع بيانات، مثل الأرقام أو النصوص.

تتميز المصفوفات بقدرتها على تخزين كميات كبيرة من البيانات في مكان واحد، مما يجعل الوصول إليها وإجراء العمليات عليها بشكل أسهل

تقدم مكتبات مثل NumPy وظائف متقدمة لمعالجة المصفوفات، مما يسهل التعامل مع البيانات الكبيرة والمعقدة

الدوال وانواعها

الدوال في لغة جافا هي وحدات برمجية قابلة لإعادة الاستخدام تؤدي مهمة معينة.

الدوال في بايثون هي مجموعة من التعليمات التي يمكن استدعاؤها عدة مرات في البرنامج، مما يوفر وقتًا وجهدًا.

يمكن تعريف دوال عامة باستخدام def أو استخدام دوال جاهزة مثل ()print

كما يمكن للدوال أن تأخذ معاملات وتعيد نتائج.

هناك أيضًا دوال مجهولة (lambda functions) التي تستخدم لتعريف دوال قصيرة ومؤقتة.

استخدام الدوال يساهم في تنظيم الكود وجعله أكثر قابلية لإعادة الاستخدام والصيانة

التعامل مع الفئات

الفئات (Classes) في بايثون هي القالب الذي يمكن من خلاله إنشاء كائنات

تحتوي الفئات على خصائص وطرق تمكن من تنظيم البيانات والوظائف المرتبطة بها.

يستخدم مفهوم الفئات لتطبيق البرمجة كائنية التوجه (OOP) مما يسمح بإنشاء هياكل بيانات مخصصة تتناسب مع احتياجات البرنامج.

يوفر التعامل مع الفئات مرونة وقوة أكبر في التعامل مع البيانات المعقدة وطرق تنظيمها.

مدخل إلى البرمجة كائنية التوجه

البرمجة كائنية التوجه (Object-Oriented Programming - OOP) هي نموذج برمجي يركز على تنظيم الكود حول الكائنات، التي تمثل كيانات من العالم الحقيقي أو المجرد.

تعتمد (OOP) على أربعة مبادئ أساسية:

التغليف (Encapsulation) :جمع البيانات والطرق التي تتعامل معها في كيان واحد يسمى الكائن، مما يحمي البيانات من التعديل المباشر.
الوراثة (Inheritance) : تتيح لفئة ما أن ترث خصائص وسلوكيات من فئة أخرى، مما يُعيد استخدام الكود ويوفر هيكلية تنظيمية.
التعددية (Polymorphism) : السماح للكائنات بتنفيذ نفس الطريقة بطرق مختلفة، اعتمادًا على الفئة التي تنتمي إليها.
التجريد (Abstraction) : إخفاء التفاصيل المعقدة للكائن والتركيز فقط على الجوانب المهمة.

تساعد (OOP) في كتابة كود منظم، قابل للتوسع، وسهل الصيانة.

تدعم بايثون البرمجة كائنية التوجه بشكل كامل، مما يجعلها مناسبة لتطوير تطبيقات كبيرة وقابلة للتطوير