Python 3 - introduction

DAMIAN KURPIEWSKI

Python 3

Scripting language

Interpreted, not compiled

No strong (static) typing, but types do matter

Code is not strongly structurized

Python 3 - libriaries

There are many different libraries available:

SCIENTIFIC

ARTIFICIAL INTELLIGENCE

GAME DEVELOPMENT

MOBILE APP DEVELOPMENT

AND MUCH MORE…

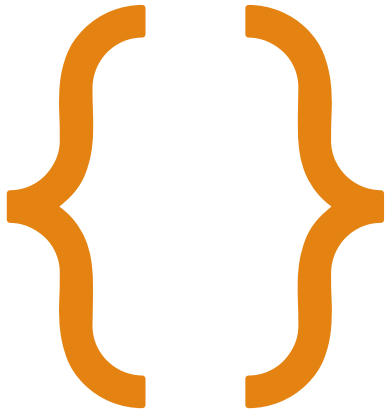
Python 3 – dependency management

To manage dependencies and automatically install and update libraries, use the **pip** command

With one command we can install the necessary library:

pip install [library name]

The required libraries for the project are usually placed in the **requirements.txt** file



Python 3

LANGUAGE CONSTRUCTIONS

Comments

```
# standard comment
```

```
"""
documentation
comment
"""
```

Variables

number = 34

text = **"Hello World"**

real_number = 3.4

Arithmetic operators

```
a = 10
b = 3

sum = a + b             # 13
product = a * b         # 30
quotient = a / b        # 3.3333...
modulo = a % b          # 1
power = a ** b          # 1000
```

Assignment operators

```
a = 10    # 10
a += 1    # 11
a -= 1    # 10
a *= 2    # 20
a /= 2    # 10.0
a %= 4    # 2.0
a **= 3   # 8.0
```

Comparison operators

```
a = 10
b = 5

a == b    # False
a != b    # True
a < b     # False
a <= b    # False
a > b     # True
a >= b    # True
```

Input

```
text = input("Input text")
```

```
number = int(input("Input number"))
```

Output

```
print("Hello World!")

a = 10
print("a = " + str(a))
print(f"a = {a}")
```

Conditional statement

```
temperature = 25

if temperature < 10:
    print("Cold!")
elif temperature < 20:
    print("Warm!")
else:
    print("Hot!")
```

Conditional while loop

```
x = 0
while x < 10:
    print(f"x = {x}")
    x += 1
```

For counting loop

```
for i in range(0, 10):
    print(f"i = {i}")
```

For counting loop with negative step

```
for i in range(10, 0, -1):
    print(f"i = {i}")
```

Lists

```
empty_list = []
print(len(empty_list))   # 0

filled_list = [1, 2, 3, 4, 5]
print(len(filled_list))   # 5
```

Lists

```
lst = []
print(len(lst))   # 0

lst.append(10)
print(len(lst))   # 1

lst.append(20)
lst.append(30)
print(len(lst))   # 3
print(lst)        # [10, 20, 30]

print(lst[0])     # 10
print(lst[1])     # 20
print(lst[2])     # 30

print(lst[-1])    # 30
print(lst[-2])    # 20
```

Functions

```
def sum(a, b):
    return a + b


print(sum(2, 5))   # 7
```

Functions

```
def point(a):
    x = a
    y = a * 2
    return x, y


print(point(5))   # (5, 10)
```

Functions – types suggestions

```
def product(a: int, b: int) -> int:
    return a * b
```

Classes

```
class Rectangle:

    def __init__(self, height, width):
        self.h = height
        self.w = width

    def area(self):
        return self.h * self.w

    def is_square(self):
        return self.h == self.w


rect = Rectangle(5, 10)
print(rect.area())
print(rect.is_square())

rect.h = 10
print(rect.is_square())
```