Python Basics

Variables

You can name a variable anything as long as it obeys the following rules:

1. It can be only one word.
2. It can use only letters, numbers, and the underscore (_) character.
3. It can't begin with a number.
4. Variable name starting with an underscore (_) are considered as "unuseful`.

Example:

```
spam = 'Hello'
_spam = 'Hello'
```

Math Operators

Operators	Operation	Example
**	Exponent	2 ** 3 = 8
%	Modulus/Remainder	22 % 8 = 6
//	Integer division	22 // 8 = 2
/	Division	22 / 8 = 2.75
*	Multiplication	3 * 3 = 9
-	Subtraction	5 - 2 = 3
+	Addition	2 + 2 = 4

Comments

Inline comment:

```
# This is a comment
```

Multiline comment:

```
# This is a
# multiline comment
```

Code with a comment:

```
a = 1  # initialization
```

Please note the two spaces in front of the comment.

Function docstring:

```
def foo():
    """ This is a function docstring
    You can also use:
    ''' Function Docstring '''
    """
```

Data types

Data Type	Examples
Integers	-2, -1, 0, 1, 2, 3, 4, 5
Floating-point numbers	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', 'W0rld'

Co-funded by the Erasmus+ Programme of the European Union

Python Basics

Variables

You can name a variable anything as long as it obeys the following rules:

1. It can be only one word.
2. It can use only letters, numbers, and the underscore (_) character.
3. It can't begin with a number.
4. Variable name starting with an underscore (_) are considered as "unuseful`.

Example:

```
spam = 'Hello'
_spam = 'Hello'
```

Math Operators

Operators	Operation	Example
**	Exponent	2 ** 3 = 8
%	Modulus/Remainder	22 % 8 = 6
//	Integer division	22 // 8 = 2
/	Division	22 / 8 = 2.75
*	Multiplication	3 * 3 = 9
-	Subtraction	5 - 2 = 3
+	Addition	2 + 2 = 4

Comments

Inline comment:

```
# This is a comment
```

Multiline comment:

```
# This is a
# multiline comment
```

Code with a comment:

```
a = 1  # initialization
```

Please note the two spaces in front of the comment.

Function docstring:

```
def foo():
    """ This is a function docstring
    You can also use:
    ''' Function Docstring '''
    """
```

Data types

Data Type	Examples
Integers	-2, -1, 0, 1, 2, 3, 4, 5
Floating-point numbers	-1.25, -1.0, --0.5, 0.0, 0.5, 1.0, 1.25
Strings	'a', 'aa', 'aaa', 'Hello!', 'W0rld'

Co-funded by the Erasmus+ Programme of the European Union

https://teducativas.madeira.gov.pt/roboloco/

Python Basics

Conditions

One block:

```
x = 3
if x == 3:
    print('x is 3')
```

Two blocks:

```
mark = 80
if mark >= 50:
    print('pass')
else:
    print('fail')
```

Multiple blocks:

```
mark = 80
if mark >= 65:
    print('credit')
elseif mark >= 50:
    print('pass')
else:
    print('fail')
```

Range

Counts from 0 to 9:

```
range(10)
```

Starts from 0 and goes up to, not including 10

Counts from 1 to 10:

```
range(1, 11)
```

Counts from 10 to 1:

```
range(10, 0, -1)
```

Counts in steps of 2:

```
range(0, 11, 2)
```

Counts down in steps of 2:

```
range(10, 0, -2)
```

Loops

While:

```
x = 0
while x < 4:
    print(x)
    x = x + 1
```

Exiting a loop using break:

```
x = 0
while x < 4:
    print(x)
    if x == 2:
        break
    x = x + 1
```

Restart the loop using continue:

```
x = 0
while x <= 10:
    x = x + 1
    if x % 2 == 0:
        continue
    print('%s is odd' % x)
```

For:

```
for i in range(10):
    print(i)
```

Iterates over string:

```
for c in 'Hello':
    print(c)
```

Comparison

Equal to:

```
x == 3:
```

Not equal to:

```
x != 3:
```

Less than:

```
x < 3:
```

Greater than:

```
x > 3:
```

Less than or equal to:

```
x <= 3:
```

Less than or equal to:

```
x >= 3:
```

The result of a comparison is a boolean:

```
True
```

or

```
False
```

Input and output

Print message:

```
print('Hello world!')
```

Print multiple values:

```
ndays = 365
print('There are', ndays, 'in a year')
```

Ask user for a string:

```
name= input('What is your name? ')
```

Ask user for a whole number:

```
num = int(input('Enter a number: '))
```

Co-funded by the Erasmus+ Programme of the European Union

Python Basics

Conditions

One block:

```
x = 3
if x == 3:
    print('x is 3')
```

Two blocks:

```
mark = 80
if mark >= 50:
    print('pass')
else:
    print('fail')
```

Multiple blocks:

```
mark = 80
if mark >= 65:
    print('credit')
elseif mark >= 50:
    print('pass')
else:
    print('fail')
```

Range

Counts from 0 to 9:

```
range(10)
```

Starts from 0 and goes up to, not including 10

Counts from 1 to 10:

```
range(1, 11)
```

Counts from 10 to 1:

```
range(10, 0, -1)
```

Counts in steps of 2:

```
range(0, 11, 2)
```

Counts down in steps of 2:

```
range(10, 0, -2)
```

Loops

While:

```
x = 0
while x < 4:
    print(x)
    x = x + 1
```

Exiting a loop using break:

```
x = 0
while x < 4:
    print(x)
    if x == 2:
        break
    x = x + 1
```

Restart the loop using continue:

```
x = 0
while x <= 10:
    x = x + 1
    if x % 2 == 0:
        continue
    print('%s is odd' % x)
```

For:

```
for i in range(10):
    print(i)
```

Iterates over string:

```
for c in 'Hello':
    print(c)
```

Comparison

Equal to:

```
x == 3:
```

Not equal to:

```
x != 3:
```

Less than:

```
x < 3:
```

Greater than:

```
x > 3:
```

Less than or equal to:

```
x <= 3:
```

Less than or equal to:

```
x >= 3:
```

The result of a comparison is a boolean:

```
True
```

or

```
False
```

Input and output

Print message:

```
print('Hello world!')
```

Print multiple values:

```
ndays = 365
print('There are', ndays, 'in a year')
```

Ask user for a string:

```
name= input('What is your name? ')
```

Ask user for a whole number:

```
num = int(input('Enter a number: '))
```

Co-funded by the Erasmus+ Programme of the European Union

https://teducativas.madeira.gov.pt/roboloco/