



School Nova

Computer Science 101

Class #3:

- 1) Basic data types (string, integer, float, Boolean)**
- 2) Four types of operators**
- 3) f-strings**

By Oleg Smirnov

Python Data Types, p1



name = "John Smith"	String
year = 2019	Integer
numPi = 3.14159	Float
numPi = "3.14159"	String
result = True result = False	Boolean
result = "True"	String

`type(varname)` will show the type of the variable `varname`



Python Data Types, p2

<code>str()</code>	Convert to string Almost anything can be converted Cannot convert variables which are not defined: <code>str(a)</code>, <code>a</code> not defined Cannot convert math expressions which yield an error: <code>str(10/0)</code>
<code>bool()</code>	Converts to a Boolean, which is either True or False Most variables are seen as True, with a few exceptions such as <code>bool(0)</code>



Python Data Types, p3

int()	Convert to integer Can convert strings without decimal part: <code>int('5')</code> Cannot convert strings with decimal part: <code>int('5.5')</code> is an error Cannot convert strings that do not look like an integer: <code>int('ok')</code> error Can convert a Boolean and float (rounding down): <code>int(True)</code>, <code>int(5.5)</code>
float()	Convert to a floating point number (number with a decimal part) Can convert a string that looks like a number: <code>float('5.5')</code> Can convert a Boolean and integer: <code>float(True)</code>, <code>float(5)</code>



Python Data Types, p4

String + String String * Integer String * Bool	String
Integer + Integer Integer * Integer	Integer
Integer / Integer (even if no remainder) Integer + Float Integer * Float	Float



Python Data Types, p5

String + Integer String + Float String + Bool String * Float String / Integer	Errors
Bool + Integer Bool + Float	Integer Float

Arithmetic operators



Addition	+	
Subtraction	-	
Multiplication	*	
Division	/	
Floor division	//	17 // 5 (returns 3, removes decimal)
Modulus:	%	17 % 5 (returns 2, remainder)
Exponent:	**	2 ** 3 (returns 8, 2 to the power of 3)

Comparison operators



Operator	General Form	Returns True	Returns False
<code>==</code>	<code>a == b</code>	<code>4 == 2 * 2</code>	<code>4 == 3</code>
<code>!=</code>	<code>a != b</code>	<code>4 != 3</code>	<code>4 != 2 * 2</code>
<code>></code>	<code>a > b</code>	<code>4 > 3</code>	<code>3 > 4</code>
<code><</code>	<code>a < b</code>	<code>3 < 4</code>	<code>4 < 3</code>
<code>>=</code>	<code>a >= b</code>	<code>4 >= 4</code>	<code>4 >= 5</code>
<code><=</code>	<code>a <= b</code>	<code>4 <= 4</code>	<code>4 <= 3</code>

Logical operators



and	True if both arguments are true	x and y
or	True if either of the arguments is true	x or y
not	True if the argument is false	not x

Examples (=> Python output)

False and True => False	1 < 2 and 1 == 1 => True
False or True => True	1 < 2 and 1 == 2 => False
not True => False	1 < 2 or 2 < 1 => True
not False => True	1 > 2 or 2 < 1 => False
	not 2 > 1 => False
	not 1 > 2 => True

Membership operators (first look)



in Returns True if a sequence with the specified value is present in the object

not in Returns True if a sequence with the specified value is not present in the object

The object must be **iterable**! We will study those types of objects in the future. At the moment, you know only one iterable object: **string** (it consists of characters).

`x = "School Nova"`

`y = "Nova"`

`y in x` will return **True**

`x in y` will return **False**

print() and f-strings



f-strings were first implemented in Python 3.6

```
name, age = "Sonya", 12
```

```
print(f"Meet {name}. She is {age} years old.")
```

```
print(f"Meet {name}. " \
```

```
    f"She is {age} years old.")
```

```
print(f"Meet {name}. \nShe is {age} years old.")
```