

Types, Statements and Other Goodies



Bo Milanovich

PYTHON WIZARD

@pythonbo pythonbo.com



Module Overview



Data Types

Flow Control

Loops

Dictionaries

Exceptions

Other Data Types



Types in Python – Wait, What?



Python vs. the Others... again

// C# or Java

```
int answer = 42;
```

```
String name = "PythonBo";
```

Python

```
answer = 42
```

```
name = "PythonBo"
```



Combining Data Types

```
Python 3.6.0 Shell
Python 3.6.0 (default, Feb 24 2017, 11:58:14)
[GCC 4.2.1 Compatible Apple LLVM 8.0.0 (clang-800.0.42.1)] on darwin
Type "copyright", "credits" or "license()" for more information.
>>> def add_numbers(a, b):
        print(a + b)

>>> add_numbers(5, 11)
16
>>> add_numbers(5, "something")
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    add_numbers(5, "something")
  File "<pyshell#2>", line 2, in add_numbers
    print(a + b)
TypeError: unsupported operand type(s) for +: 'int' and 'str'
>>>
```



Type Hinting

```
def add_numbers(a: int, b: int) -> int:  
    return a + b
```



Integers and Floats

```
answer = 42
pi = 3.14159
answer + pi = 45.14159    # Don't worry about conversion!
int(pi) == 3
float(answer) == 42.0
```



Strings

```
'Hello World' == "Hello World" == """Hello World"""  
"hello".capitalize() == "Hello"  
"hello".replace("e", "a") == "hallo"  
"hello".isalpha() == True  
"123".isdigit() == True # Useful when converting to int  
"some,csv,values".split(",") == ["some", "csv", "values"]
```



String Format Function

```
name = "PythonBo"
```

```
machine = "HAL"
```

```
"Nice to meet you {0}. I am {1}".format(name, machine)
```

```
f"Nice to meet you {name}. I am {machine}"
```



Boolean and None

```
python_course = True
java_course = False
int(python_course) == 1
int(java_course) == 0
str(python_course) == "True"

aliens_found = None
```



If Statements

```
number = 5
if number == 5:
    print("Number is 5")
else:
    print("Number is NOT 5")
```



Truthy and Falsy Values

```
number = 5
```

```
if number:
```

```
    print("Number is defined and truthy")
```

```
text = "Python"
```

```
if text:
```

```
    print("Text is defined and truthy")
```



Boolean and None

```
python_course = True  
  
if python_course: # Not python_course == True  
    print("This will execute")
```

```
aliens_found = None  
  
if aliens_found:  
    print("This will NOT execute")
```



Not If

```
number = 5
```

```
if number != 5:
```

```
    print("This will not execute")
```

```
python_course = True
```

```
if not python_course:
```

```
    print("This will also not execute")
```



Multiple If Conditions

```
number = 3
```

```
python_course = True
```

```
if number == 3 and python_course:  
    print("This will execute")
```

```
if number == 17 or python_course:  
    print("This will also execute")
```



Ternary If Statements

```
a = 1
```

```
b = 2
```

```
"bigger" if a > b else "smaller"
```



Lists

```
student_names = []  
student_names = ["Mark", "Katarina", "Jessica"]
```



Getting List Elements

```
student_names = ["Mark", "Katarina", "Jessica"]
```

```
student_names[0] == "Mark"
```

```
student_names[2] == "Jessica"
```

```
student_names[-1] == "Jessica"
```



Changing List Elements

```
student_names = ["Mark", "Katarina", "Jessica"]  
student_names[0] = "James"  
student_names == ["James", "Katarina", "Jessica"]
```



List Functions

```
student_names = ["Mark", "Katarina", "Jessica"]  
student_names[3] = "Homer"    # No can do!  
student_names.append("Homer")  # Add to the end  
student_names == ["Mark", "Katarina", "Jessica", "Homer"]  
"Mark" in student_names == True  # Mark is still there!  
len(student_list) == 4    # How many elements in the list  
del student_names[2]    # Jessica is no longer in the list :(  
student_names = ["Mark", "Katarina", "Homer"]
```



List Slicing

```
student_names = ["Mark", "Katarina", "Homer"]
```

```
student_names[1:] == ["Katarina", "Homer"]
```

```
student_names[1:-1] == ["Katarina"]
```



Demo



Break and Continue



While Loops

```
x = 0
while x < 10:
    print("Count is {}".format(x))
    x += 1
```



Infinite Loops

```
num = 10
while True:
    if num == 42:
        break
    print("Hello World")
```



Dictionaries

```
student = {  
    "name": "Mark",  
    "student_id": 15163,  
    "feedback": None  
}
```



List of Dictionaries

```
all_students = [  
    {"name": "Mark", "student_id": 15163 },  
    {"name": "Katarina", "student_id": 63112 },  
    {"name": "Jessica", "student_id": 30021 }  
]
```



Dictionary Data

```
student["name"] == "Mark"
```

```
student["last_name"] == KeyError
```

```
student.get("last_name", "Unknown") == "Unknown"
```

```
student.keys() = ["name", "student_id", "feedback"]
```

```
student.values() = ["Mark", 15163, None]
```

```
student["name"] = "James"
```

```
del student["name"]
```



Demo



Exceptions



Other Data Types

complex

long # Only in Python 2

bytes and bytearray

tuple = (3, 5, 1, "Mark")

set and frozenset

set([3, 2, 3, 1, 5]) == (1, 2, 3, 5)



Summary



Data types

Lists

Dictionaries

For and while loops

Exception handling

