Back to Basics

Gaurav Pandey

PyCon Italia 2022, Florence, Italy

The Zen of Python

Beautiful is better than ugly.
Explicit is better than implicit.
Simple is better than complex.
Complex is better than complicated.
Flat is better than nested.
Sparse is better than dense.
Readability counts.
Special cases aren't special enough to break the rules.
Although practicality beats purity.
Errors should never pass silently.
Unless explicitly silenced.
In the face of ambiguity, refuse the temptation to guess.
There should be one-- and preferably only one --obvious way to do it.
Although that way may not be obvious at first unless you're Dutch.
Now is better than never. Although never is often better than *right* now.
If the implementation is hard to explain, it's a bad idea.
If the implementation is easy to explain, it may be a good idea.
Namespaces are one honking great idea -- let's do more of those!

About Me

Just a Python Developer

@yednapg

Let’s Go

https://bit.ly/gaurav-pycon-italia-22-slides

Topics for Discussion

- Variables, and Data Types
- Conditionals (if, else, and elif)
- Iteration (for, and while loop)
- Tips and Tricks, Best Practices
- Q/A

Variables

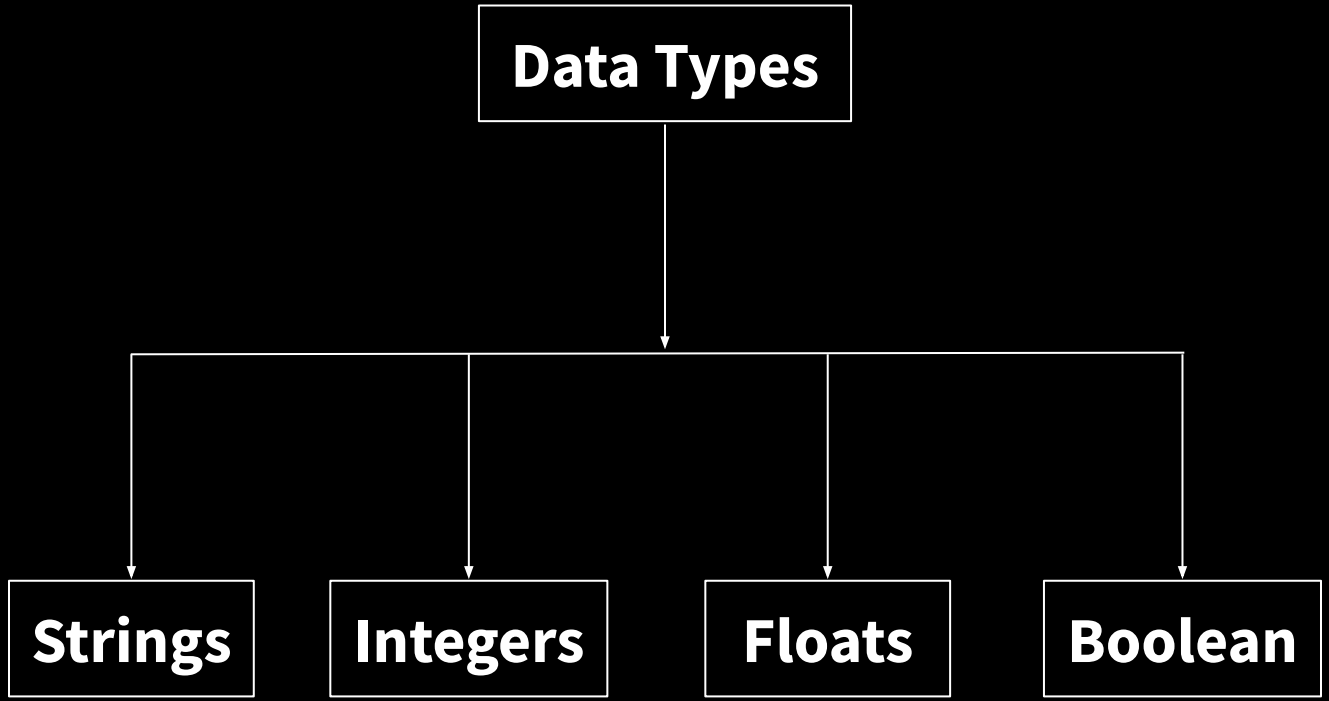
Variables

Variables are a name that refers to a value or we can say that variables are container for storing data values/types.

For Example;

```
>>> a = “hi,how are you?”
>>> first_var = 7.9
>>> no_45 = 7
>>> A = “hello”
```

Data Types


Data Types
Strings
Integers
Floats
Boolean

Strings

Strings

Text in Python, is represented as *str* or STRINGS. They can be enclosed by either double or single quotes

For Example;

```
>>> a = “hi,how are you?”
>>> type(a)
<class ‘str’>
```

Quiz

Strings Quiz

Question: What is the output of the given Code?

```
>>> a = “nearby”
>>> b = “tea”
>>> print(‘byte’ in 3*(a + b))
```

Answer

Strings Quiz, Answer

Question: What is the output of the given Code?

```
>>> a = “nearby”
>>> b = “tea”
>>> print(‘byte’ in 3*(a + b))
True
```

How?

Strings Quiz, How?

Question: What is the output of the given Code?

```
>>> a = “nearby”
>>> b = “tea”
>>> print(‘byte’ in 3*(a + b))
True

# nearbyteanearbyteanearbytea
```

Integers & Floats

Integers, and Floats

Integers or Int are Zero, Positive or Negative Whole Numbers without any fractions and Floats or float are a numbers having one or more decimal in it.

For Example;

```
>>> a = 7
>>> b = 3.14
>>> print(type(a))
<class ‘int’>
>>> print(type(b))
<class ‘float’>
```

Quiz

Integers, and Floats Quiz

Question: What is the output of these statements?

```
>>> print(5 + (12.0 - 7.0))
>>> print(6 + 5)
>>> print(type('12'))
```

Answer

Integers, and Floats Quiz, Answer

Question: What is the output of these statements?

```
 >>> print(5 + (12.0 - 7.0))
10.0
 >>> print(6 + 5)
11
 >>> print(type('12'))
<class 'str'>
```

Boolean

Boolean

Boolean or bool Data Type in Python has only two values i.e. True or False. They tell whether the expression is True or False.

For Example;

```
>>> a = 7
>>> a == int
True
>>> a != float
True
>>> a == str
False
```

Quiz

Boolean Quiz

Question: What is the output of these statements?

For Example;

```
>>> a = 7
>>> a == int
>>> a != float
```

Answer

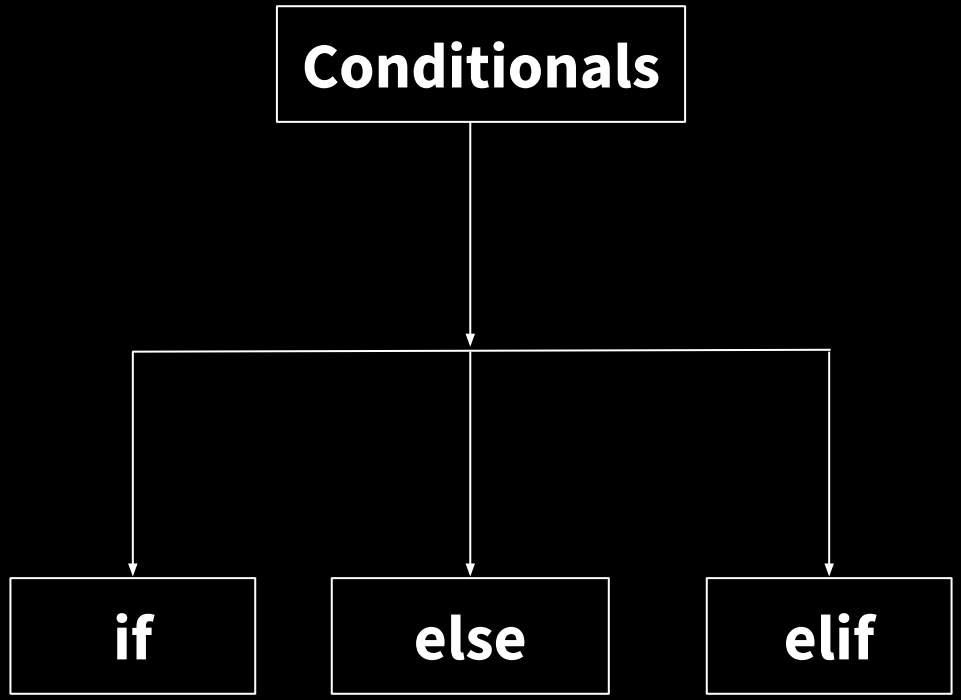
Boolean Quiz, Answer

Question: What is the output of these statements?

For Example;

```
>>> a = 7
>>> a == int
True
>>> a != float
True
```

Conditionals


Conditionals
if
else
elif

if statement

if statement

```
>>> a = 10
>>> b = 7
>>> if a > b:
        print(“a is greater than b”)

# a is greater than b
```

else statement

else statement

```
>>> a = 10
>>> b = 7
>>> if a > b:
        print(“a is greater than b”)
    else:
        print(“Please Try Again”)

# a is greater than b
```

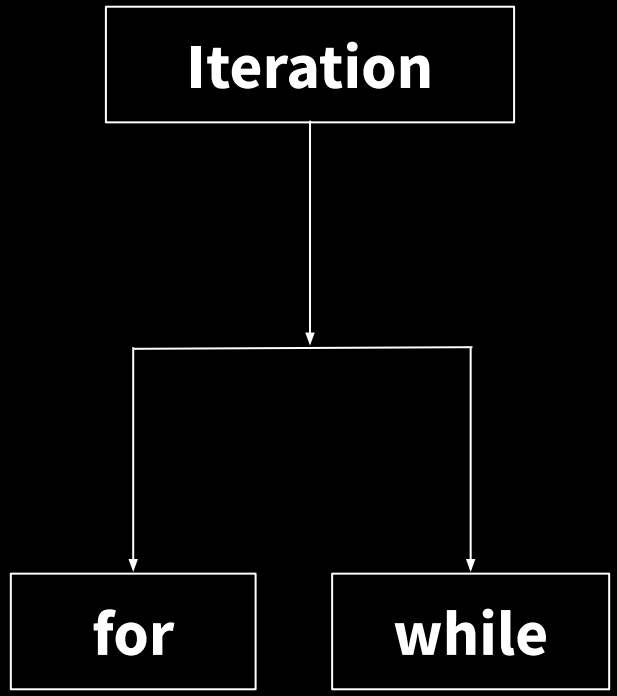
elif statement

elif statement

```
>>> a = 10
>>> b = 7
>>> if a > b:
        print(“a is greater than b”)
    elif a == b:
        print(“both are equal”)

# a is greater than b
```

Iteration


Iteration
for
while

while loop

while loop

```
>>> n = 10
>>> while n < 20:
        print(n)
        n = n + 10
    print(“Done”)

# 10
# Done
```

for loop

for loop

```
>>> a = [“Florence”, “Paris”, “London”]
>>> for florence in a:
        print(“I love Florence!”)
        Break

# I love Florence!
```

Tips & Tricks

Return Multiple values from Functions

```
>>> def x():
    return 1,2,3,4

>>> a,b,c,d = x()
print(a,b,c,d)

# 1,2,3,4
```

Splitting a single String to List

```
>>> s = “a,b,c”
>>> print(s.split(,))

# [“a”, “b”, “c”]
```

F-Strings

```
>>> import datetime
>>> today = datetime.datetime.today()
>>> print(f”{today:%B %d , %Y}”)

# June 4, 2022
```

Swapping of two Numbers

```
>>> a = 5
>>> b = 7
>>> a ,b = b, a
>>> print(a, b)

# 7, 5
```

Question for You?

Why Python is Called Python?

100+ Tips and Tricks in Python

https://github.com/yednapg/pytips

Thank You

@yednapg

PyCon Italia 2022, Florence, Italy

Any Questions?