Python Cheat Sheet

Mosh Hamedani

Variables

We use variables to temporarily store data in computer’s memory.

```
price = 10

rating = 4.9

course_name = ‘Python for Beginners’

is_published = True
```

In the above example,

- **price** is an *integer* (a whole number without a decimal point)
- **rating** is a *float* (a number with a decimal point)
- **course_name** is a *string* (a sequence of characters)
- **is_published** is a *boolean*. Boolean values can be True or False.

If Statements

```
if is_hot:
   print(“hot day”)
elif is_cold:
   print(“cold day”)
else:
   print(“beautiful day”)
```

Functions

We use functions to break up our code into small chunks. These chunks are easier to read, understand and maintain. If there are bugs, it’s easier to find bugs in a small chunk than the entire program. We can also re-use these chunks.

```
def greet_user(name):
    print(f”Hi {name}”)


greet_user(“John”)
```

```
def square(number):
   return number * number

result = square(2)
print(result)  # prints 4
```

For loops

```
for i in range(1, 5):
   print(i)
```

While loops

```
i = 1
while i < 5:
   print(i)
   i += 1
```

Comments

We use comments to add notes to our code. Good comments explain the hows and whys, not what the code does. That should be reflected in the code itself. Use comments to add reminders to yourself or other developers, or also explain your assumptions and the reasons you’ve written code in a certain way.

```
# This is a comment and it won’t get executed.
# Our comments can be multiple lines.
```

Strings

We can define strings using single (‘ ’) or double (“ ”) quotes.

To define a multi-line string, we surround our string with tripe quotes (“””).

We can get individual characters in a string using square brackets [].

```
course = ‘Python for Beginners’
course[0]   # returns the first character
course[1]   # returns the second character
```

Lists

```
numbers = [1, 2, 3, 4, 5]
numbers[0]            # returns the first item
numbers[1]            # returns the second item
```

Classes define templates or blueprints for creating objects. An object is an instance of a class. Every time we create a new instance, that instance follows the structure we define using the class.

```
point1 = Point(10, 5)
point2 = Point(2, 4)
```

Note about using Classes