DIT gentle introduction to Python 2nd Edition, February 2022

2. Python 4 Poets

Alberto Barrón-Cedeño

Alma Mater Studiorum-Università di Bologna
a.barron@unibo.it @_albarron_

08/02/2022

Programming

Diagram borrowed from L. Moroney's Introduction to TensorFlow for Artificial Intelligence, Machine Learning, and Deep Learning

Conditionals

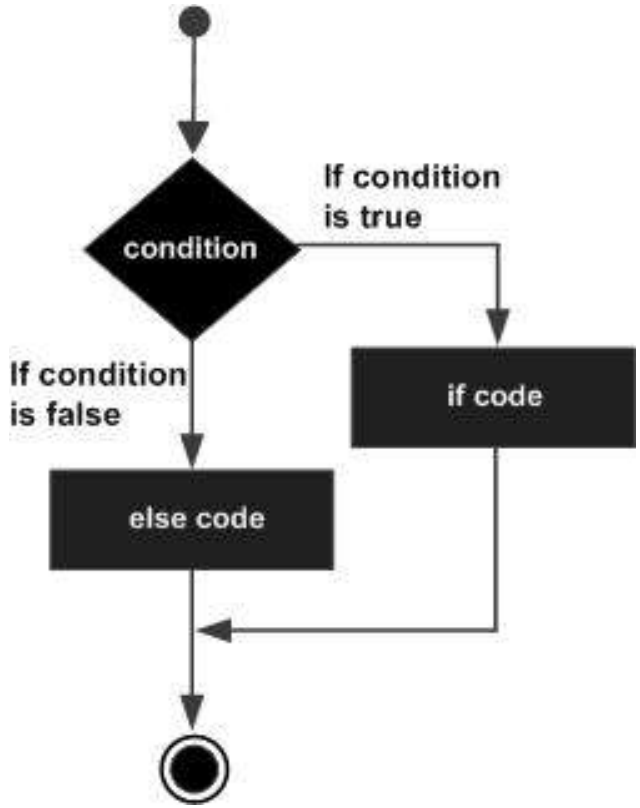

Diagram borrowed from `https://www.tutorialspoint.com`

Loops

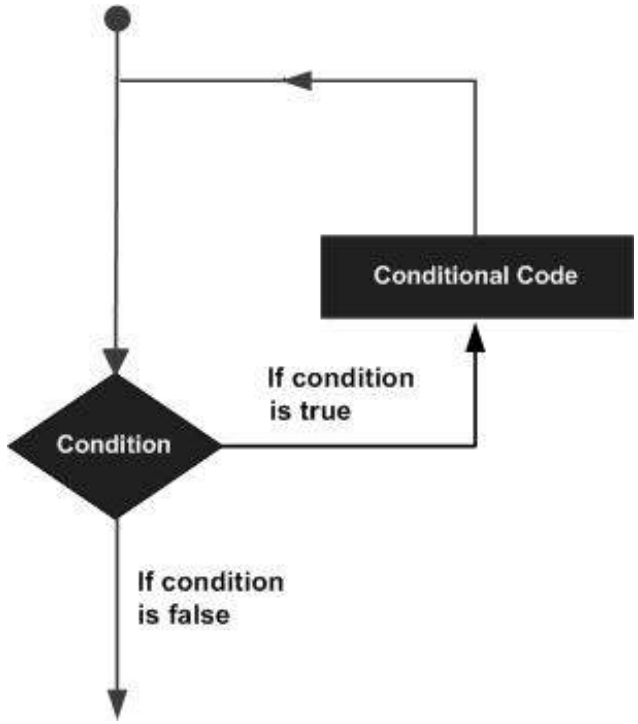

Diagram borrowed from `https://www.tutorialspoint.com`

Functions (methods)

A simple function

```
def name_of_the_function(input1, input2):
    # function code
    none
```

Functions (methods)

A simple function

```
def name_of_the_function(input1, input2):
    # function code
    none
```

Calling the function

```
name_of_the_function("hi", "ho")
```

Functions (methods)

A simple function

```
def name_of_the_function(input1, input2):
    # function code
    none
```

Calling the function

```
name_of_the_function("hi", "ho")
```

Another valid call

```
name_of_the_function(hi, ho)
```

Functions (methods)

A simple function

```
def name_of_the_function(input1, input2):
    # function code
    none
```

Calling the function

```
name_of_the_function("hi", "ho")
```

Another valid call

```
name_of_the_function(hi, ho)
```

An invalid call

```
name_of_the_function(hi)
```

A simple method to *salute* people

```
greeting_inputs = ("hey", "morning", "evening", "hi",
                   "whatsup", "hello")
greeting_responses = ["hey", "hey hows you?", "*nods*",
                   "hello, how you doing", "hello",
                   "Welcome, I am good and you"]

def generate_greeting_response(input):
    for token in input.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

Derived from

https://stackabuse.com/python-for-nlp-creating-a-rule-based-chatbot/

A simple method to *salute* people

```
greeting_inputs = ("hey", "morning", "evening", "hi",
                   "whatsup", "hello")
greeting_responses = ["hey", "hey hows you?", "*nods*",
                      "hello, how you doing", "hello",
                      "Welcome, I am good and you"]

def generate_greeting_response(input):
    for token in input.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

- greeting_inputs is a **set**
- greeting_responses is a **list**

Derived from

https://stackabuse.com/python-for-nlp-creating-a-rule-based-chatbot/

A simple method to *salute* people

```
greeting_inputs = ("hey", "morning", "evening", "hi",
                   "whatsup", "hello")
greeting_responses = ["hey", "hey hows you?", "*nods*",
                   "hello, how you doing", "hello",
                   "Welcome, I am good and you"]

def generate_greeting_response(input):
    for token in input.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

A simple method to *salute* people

```
greeting_inputs = ("hey", "morning", "evening", "hi",
                   "whatsup", "hello")
greeting_responses = ["hey", "hey hows you?", "*nods*",
                   "hello, how you doing", "hello",
                   "Welcome, I am good and you"]

def generate_greeting_response(input):
    for token in input.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

```
generate_greeting_response("hi")
```

A simple method to *salute* people

```
greeting_inputs = ("hey", "morning", "evening", "hi",
                   "whatsup", "hello")
greeting_responses = ["hey", "hey hows you?", "*nods*",
                   "hello, how you doing", "hello",
                   "Welcome, I am good and you"]

def generate_greeting_response(input):
    for token in input.split():
        if token.lower() in greeting_inputs:
            return random.choice(greeting_responses)
```

```
generate_greeting_response("hi")
```

```
generate_greeting_response("ciao")
```

From Unix to Python

- Kenneth W. Church's **Unix for poets**[1]

[1] `https://web.stanford.edu/class/cs124/kwc-unix-for-poets.pdf`

From Unix to Python

- Kenneth W. Church's **Unix for poets**[1]

Python for Poets

[1] `https://web.stanford.edu/class/cs124/kwc-unix-for-poets.pdf`

From Unix to Python

- Kenneth W. Church's **Unix for poets**[1]

↓

Python for Poets

`shorturl.at/stU36`

[1] `https://web.stanford.edu/class/cs124/kwc-unix-for-poets.pdf`