DIT gentle introduction to Python
1st Edition, November 2020

2. Python 4 Poets

Alberto Barrón-Cedeño

Alma Mater Studiorum-Università di Bologna
a.barron@unibo.it   @_albarron_
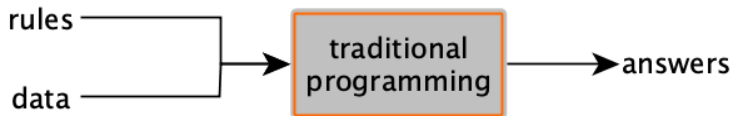
18/11/2020

Programming

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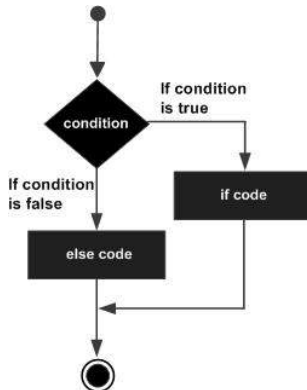
Conditionals

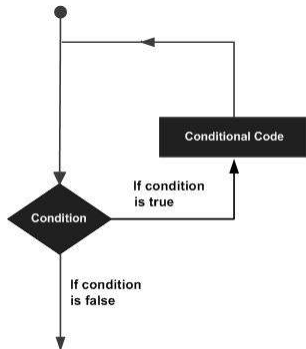Diagram borrowed from https://www.tutorialspoint.com

Loops

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

Functions (methods)

```
def name_of_the_function(input1, input2):
    function code
```

Functions (methods)

```
def name_of_the_function(input1, input2):
    function code
```

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

Functions (methods)

```
def name_of_the_function(input1, input2):
    function code
```

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

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

A simple method to *salute* people

```python
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

```python
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

```python
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

```python
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)
```

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

A simple method to *salute* people

```python
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)
```

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

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

From Unix to Python

- Kenneth W. Church's **Unix for poets**

From Unix to Python

- Kenneth W. Church's **Unix for poets**

$\downarrow$

Python for Poets

From Unix to Python

- Kenneth W. Church's **Unix for poets**

$$\downarrow$$

Python for Poets

`shorturl.at/stU36`