UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Lecture 3:
Functions and Loops

Computing In The News

Face Recognition Struggles to Recognize Us After Five Years of Aging
New Scientist
Matthew Sparkes August 24, 2022

A test designed by the Norwegian University of Science and Technology's Marcel Grimmer and colleagues found that facial recognition algorithms start running into difficulty identifying people after they have aged five years. The researchers used open-source alternatives to face recognition tools used by police and smartphone manufacturers, as well as artificial intelligence-generated images of 50,000 humans aged synthetically. Grimmer said the tools' accuracy declined continuously from the point the reference image was captured. The algorithms used to age faces synthetically from reference images also proved more effective when the target was between 20 and 40 years, compared to children and older adults. The implication is that new photos may be needed more often to maintain accuracy and security.

Announcements

- We are working to expand the course!
  - Next week the waitlist will start to clear.
  - At least 40 seats on the waitlist
  - All concurrent enrollment students will get in (a separate waitlist)
- We'll be adding 1-3 new lab sections
  - Times and signup info next week when we figure out the schedules.

Let's talk Python

- Expression `3.1 * 2.6`
- *Call* expression `max(0, x)`
- Variables `my_name`
- Assignment Statement `my_name = <expression>`
- Define Statement: `def function_name(<arguments>):`
- Control Statements:
  `if …`
  `for …`
  `while …`
- Comments `# Text after the # is ignored.`

UC Berkeley EECS
Lecturer
Michael Ball

Python: Control Flow

Conditional Statement

- Do some statements, conditional on a *predicate* expression

```
if <predicate>:
        <true statements>
else:
        <false statements>
```

- Example:

```
if temperature > 98.6:
        print("fever!")
else:
        print("no fever")
```

Live Coding Demo

```
course = 'C88C'
time = '1:00'
if time == '1:00':
    print("Go to " + course)
else:
    print("Go to some other class")


Go to C88C
```

UC Berkeley EECS
Lecturer
Michael Ball

Python: Definitions and Control

Learning Objectives

- Create your own functions.
- Write a loop to run the same code multiple times
- Use conditionals to control when a loop stops

Functions in Python

- We "define" them with `def`
- We typically `name_them_using_underscores` ("Snake case")
- The first line ends in a `:`
- The body is indented by 4 spaces
- Arguments (parameters) create 'names' that exist only in our function
- Most functions will return a value, but some do not.

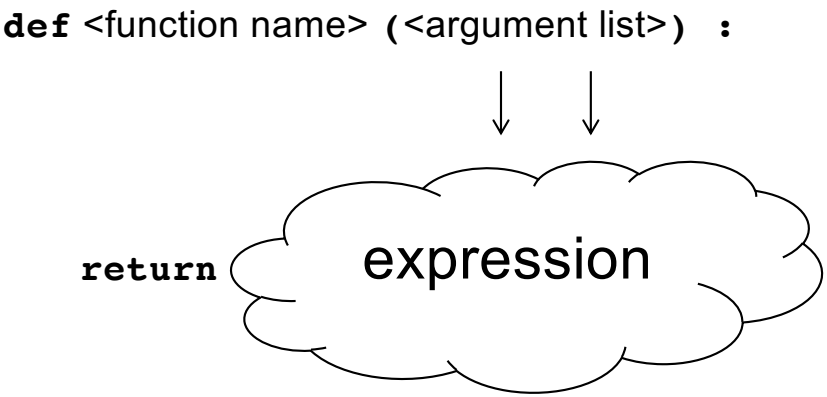
```
def greet(name):
    print("Hello, " + name)
```

Functions: Example

```
>>> y  = 5
>>> x = 3
>>> z = max(3, 5) * 10
>>> z
50
```

```
def max(x, y):
    if x > y:
        return x
    else:
        return y
```

Defining Functions



- Abstracts an expression or set of statements to apply to lots of instances of the problem
- A function should *do one thing well*

Returns and Values

- All functions always return SOME value.
- If you don't specify `return`, the value is `None`.

Functions: Calling and Returning Results

Python Tutor

```
# This style is shorthand.
def max(x, y):
    return x if x > y else y

x = 3
y = 4 + max(17, x + 6) * 0.1
z = x / y
```

Doctests

- Write the docstring to explain *what* it does
  - What does the function return? What are corner cases for parameters?

```
def max(x, y):
    """Returns the larger value of arguments x and y
    >>> max(6, 0)
    6
    """
    return x if x > y else y
```

- Write doctest to show what it should do
  - Before you write the implementation.
  - python3 –m doctest [-v] file.py

UC Berkeley EECS
Lecturer
Michael Ball

Computational Structures in Data Science

Iteration with `while` Loops

Learning Objectives

- Use a while loop to repeat some task.
- Write an expression to control when a while loop stops executing

`while` Statement – Iteration Control

- Repeat a block of statements until a predicate expression is satisfied

```
<initialization statements>
while <predicate expression>:
    <body statements>

<rest of the program>
```

Sum The Numbers

- This is a task we'll see many times!

```
total = 0
n = 1
while n <= 10:
    total += n
    n += 1
print(total)
```

UC Berkeley EECS
Lecturer
Michael Ball

Iteration With `for` Loops

Learning Objectives

- Compare a `for` loop and a `while` loop.
- Learn to use `range()`
- Use a string as a sequence of letters

`for` Statement – Iteration Control

- Repeat a block of statements for a structured sequence of variable bindings

```
<initialization statements>
for <variables> in <sequence expression>:
  <body statements>

<rest of the program>
```

`<sequence expression>` — What's that?

- Sequences are a *type* of data that can broken down into smaller parts.
- Common sequences:
  - range() – gimme all the numbers
  - strings
  - lists (next week!)
- We'll start with two basic facts:
  - range(10) is the numbers 0 to 9, or range(0, 10)
  - `[]` means "indexing" an item in a sequence.
  - `"Hello"[0] == "H"`

Data-Driven Iteration

- describe an expression to perform on each item in a sequence
- let the data dictate the control

```
[ <expr with loop var> for <loop var> in <sequence expr > ]
```