Computational Structures in Data Science

Lecture 3:
Control

Conditionals and Loops

UC Berkeley

Announcements

- Lab Attendance: Autograder is still a WIP
  - But if you attended lab, filled out a code, you only need it to say 2/4
  - We're working on making it more clear very son.
- Earning points is based on *correctness*
  - You get as many tries as you need, but the results must work, at the end of the day.
  - If you need an extension, you can ask for one, but be careful with time. ☺

Computational Structures in Data Science

Learning Process & Debugging

UC Berkeley

Process NOT Memorization

- This is not a class about memorization.
- This is a class about *problem solving* and *process.*
- You will not know everything, but you will be able to figure it out.
- Focus on building intuition!
  - **Predict** what will happen **first**
  - Then **try and inspect**
  - Now, Figure out **why**!
  - Was your prediction correct or incorrect?

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

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.`

Computational Structures in Data Science

Python: Control Flow

UC Berkeley

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 = '2:00'
if time == '4:00':
    print(f"Go to {course}")
else:
    print("Go get some ☕")

Go to C88C
```

Conditional Expression Shorthand

- Return a Value Based on some condition

```
<true expression> if <predicate> else <false expression>
```

- Example:

```
status = "it's hot!" if temperature > 85 else 'not hot…'
```

Opinion: Not my favorite syntax, useful to be aware of, but write code for what is readable *to you!*

Computational Structures in Data Science

Iteration with `while` Loops

UC Berkeley

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
- The body is indented one level (usually 4 spaces, just like a function)

```
<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)
```

Sum The Numbers

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

```
total = 0
n = 1
while True:
    total += n
    n += 1
    if n > 10:
        break

print(total)
```

Computational Structures in Data Science

Iteration With `for` Loops

UC Berkeley

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>
```

Demo: Sum The Numbers with `range`

- Let's try the same task as before, but with a for loop

```
total = 0
for number in range(0, 11):
  total = total + number
print(total)
```

<sequence expression> — What's that?

- Sequences are a type of data that can broken down into smaller parts.
- Common sequences:
  - range() – give me all the numbers
  - Strings, e.g, "Hello, C88C!"
    - What is it a sequence of? Characters!
  - lists (next!)
  - tuples, dictionaries, tables (Data 8) — these will come later.
- 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 > ]
```