print(“Hello, World!”)

NCSS Challenge - Beginners
Week 3 Part 2

What will we cover?

- Iteration with for loops;
- Looping over a string
- Looping over a range of numbers;
- Comments;
- Loops with the turtle.

GROK LEARNING

What does this cover?

- Design, modify and follow simple algorithms involving sequences of steps, branching, and **iteration (repetition)** (ACTDIP019)
- Implement digital solutions as simple visual programs involving branching, **iteration (repetition)**, and user input (ACTDIP020)
- Implement... **iteration** and functions in a **general-purpose programming language** (ACTDIP030)

1 Iteration with for loops

Motivation: Iterating over the letters in a name!

- Imagine we want to read in a person’s name, and print out each letter.

```
name = input('Enter your name? ')
print(name[0])
print(name[1])
print(name[2])
print(name[3])
```

- What are some problems with this?

For loops: a way of looping over each things!

- We want a solution that will work regardless of how long a person’s name is!

```
name = input('Enter your name? ')
for character in name:
    print(character)
```

- The for loop runs a code block for each element in a sequence, in this case, character.

Teacher Aside: How does it know what a character is?

- Some students misunderstand the name of the temporary variable that is set (e.g. `for character in name`). It's worth reiterating that the name of the variable is irrelevant, it's just a placeholder.
- But then also be prepared for:

```
for swag in lol:
```

GROK LEARNING

Test it out!

Try the first question now!

2 Looping with numbers

Looping a set number of times

- We can loop a certain, set number of times:

  ```
  for i in range(3):
    print(i)
  ```

  ```
   → 0
   → 1
   → 2
  ```

- Notice that `range` starts from zero and goes up to `n - 1` (rather than from `1` to `n`)!

Looping a set number of times

- Adding an optional second argument to `range` changes the first number that we count from.

```
for n in range(3, 6):
  print(n)
```

```
 → 3
 → 4
 → 5
```

Teacher Aside! - Think outside the box!

- What would happen if…?
- Now is a good time to have students explore the `range` function, to see if they uncover something in the next few slides for themselves!

Step by step

- We can add an optional third argument to `range` to count the *step* between the numbers.

```
for n in range(3, 12, 2):
  print(n)
```

```
→ 3
→ 5
→ 7
→ 9
→ 11
```

Putting things together, and negative counting

- We can also combine `range` options together, and use negative numbers.

```
for n in range(-10, -7, 1):
  print(n)
```

```
 → -10
 → -9
 → -8
```

Taking a step backwards

- Using a negative value for the step argument in `range` is a handy way of counting backwards!

```
for n in range(5, 0, -1):
  print(n)
```

```
→ 5
→ 4
→ 3
→ 2
→ 1
```

Test it out!

Try the second question now!

3 Comment your code!

Adding comments to your code

- You can add comments to your code, - notes to remind you what your code does.
- In Python, start your comments with a hash (#)

```
# This is a comment, it does nothing.
```

- Comments are a very good habit to get into from the start!

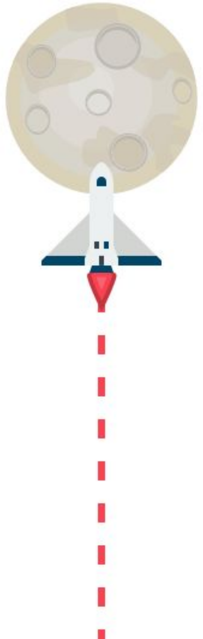
Teacher Aside!

- It’s hard to write many interesting programs without input from the user, so almost all of our questions use input.
- Using input is one of the hardest concepts for students to grasp. It often helps to act it out with a memory diagram, or writing on a piece of paper.

GROK LEARNING

3 More Python Turtle!

As normal, let’s jump in and get started!



Any Questions?

Find me at:
@groklearning
nicky@groklearning.com