print("Hello, World!")

NCSS Challenge - Beginners
Week 4 Part 2
What will we cover?

- Slices of strings;
- Making decisions inside a loop;
- Turtles and polygons;
- Angles and circles.
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. String slicing
String indexing revision

- Recall that we can index specific spots in a string:

  ```python
  msg = 'hello world'
  print(msg[0])  # → h
  print(msg[1])  # → e
  ```

- Remember that in computer science, we start counting from 0 rather than from 1!
String indexing and slicing

○ If we wanted more than a single letter, we could:

```python
msg = 'hello world'
→ wor
```

○ There's a nicer way! A piece of a string (called a **substring**) can be accessed by using two numbers separated by a colon:

```python
print(msg[6:9])
→ wor
```
String indexing and slicing

- These slices count from the first number (index) and go up to but not including the final index:

```python
msg = 'hello world'
print(msg[0:3])
→ hel
print(msg[6:10])
→ worl
```
Teacher aside - Get the class involved!

- This is a great time to get the whole class involved in guessing what will be printed out when testing out slices!
- Think outside the box. Try out slices that are too long, and that use negative numbers etc.
A common slice is from a point in a string to the end of the string. We could do this:

```python
msg = 'hello world'
print(msg[3:len(msg)])
→ lo world
```

If we leave out the second index in a slice (keeping the :!) then it will give us the substring to the end:

```python
print(msg[3:])
→ lo orld
```
Test it out!

Try the first question now!
Making decisions inside loops
Making decisions inside a loop

- Just as we nest an `if` statement inside another `if` statement, we can nest an `if` statement in a loop:

```python
line = input('Enter a line: ')
while line:
    if 'cat' in line:
        print('I see a cat!'
    else:
        print('No cat.'
    line = input('Enter a line: ')
```
Teacher Aside!

- Indenting is really important!
- Getting the right indenting level is really tricky here, so pay attention!

```python
line = input('Enter a line: ')
while line:
    if 'cat' in line:
        print('I see a cat!')
    else:
        print('No cat.')
    line = input('Enter a line: ')
```
Test it out!

Try the second questions now!
More Turtle!

Pen thickness, exterior angles, and calculating clocks!
Test it out!
Try the Turtle questions now!
Any Questions?

Find me at:
@groklearning
nicky@groklearning.com