print("Hello, World!")

NCSS Challenge - Beginners
Week 5 Part 1

GROK LEARNING

What will we cover?

- Storing lists of values;
- List methods;
- Separating strings into lists;
- Modifying a list;
- Constructing lists with loops.

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)
- Implement modular programs, **applying selected** algorithms and **data structures** including using an object-oriented programming language (ACTDIP041)

GROK LEARNING

1 Storing lists of values

GROK LEARNING

List motivation - case study!

- Jump over to the notes for a motivating example of why data structures, - e.g. lists, are useful!

Lists of (all the) things!

- You can make lists anything!
  - Strings, integers, floats, even other lists!
- Lists use square bracket notation

```
odds = [1, 3, 5, 7, 9]
colours = ['red', 'blue', 'green', 'yellow']
authors = []
```

GROK LEARNING

Lists of (all the) things!

You can use some familiar methods on lists:

- Checking the length of a list

```
colours = ['red', 'blue', 'green', 'yellow']
print(len(colours))
```

- Checking if item ‘a’ is in the list:

```
words = ['apple', 'banana', 'mango', 'kiwi fruit']
if 'mango' in words:
  print('The word "mango" is in the list!')
```

GROK LEARNING

Items in a list vs. substrings in a string

Compare checking if an item is `in` a list to if a substring `in` is a string:

```
line = 'apple banana mango kiwi fruit'
words = ['apple', 'banana', 'mango', 'kiwi fruit']
print('kiwi' in line)
      → True
print('kiwi' in words)
      → False
```

Test it out!

Try the first question now!

GROK LEARNING

2 List methods!

GROK LEARNING

Sorting and reversing

Now that we have a list data structure, we can use some handy list methods like `sort` and `reverse`:

```
pets = ['dog', 'mouse', 'fish', 'cat']
pets.sort()
print(pets)
      → ['cat', 'dog', 'fish', 'mouse']
pets.reverse()
print(pets)
      → ['mouse', 'fish', 'dog', 'cat']
```

Teacher Aside! List methods vs. string methods

- The `sort` method sorts the items in order, e.g. strings in alphabetical order and numbers in ascending order. It **modifies the original list**, called sorting in-place, rather than returning a new sorted list.

```
pets = ['dog', 'mouse', 'fish', 'cat']
incorrect = pets.sort()
print(incorrect)
        → None
print(incorrect[0])
```

GROK LEARNING

3 Constructing lists

GROK LEARNING

Using split to construct a list from input

- We can use `split` to split up a string into list elements:

```
data = input('Enter your subjects: ')
subjects = data.split()
print(subjects)
 → Enter your subjects: maths english
science
 → ['maths', 'english', 'science']
```

Printing out a list line by line

- We can use `split` to split up a string into list elements:

```
data = input('Enter your subjects: ')
subjects = data.split()
for subject in subjects:
  print(subject)
 → Enter your subjects: maths english
science
 → maths
 → english
 → science
```

Test it out!

Try the second question now!

GROK LEARNING

4 Accessing list items

GROK LEARNING

Printing out a list line by line

- An item in a list can be accessed using an *index* just like characters in a string:

```
colours = ['red', 'blue', 'green', 'yellow']
print(colours[1])
print(colours[-1])
```

→ blue

→ yellow

Teacher Aside! Test the boundaries & edge cases!

- Try out what happens if you try to access an item outside of the length of the list!

4 Modifying and creating lists

GROK LEARNING

Adding to the end of a list

- The append method adds a new item to the end of the list:

```
pets = ['dog', 'mouse', 'fish']
print(pets)
→ ['dog', 'mouse', 'fish']
pets.append('cat')
print(pets)
   → ['dog', 'mouse', 'fish', 'cat']
```

Looping with append:

```
books = []
book = input('What book are you returning? ')
while book:
    books.append(book)
    book = input('What book are you returning? ')
books.sort()
print(books)
```

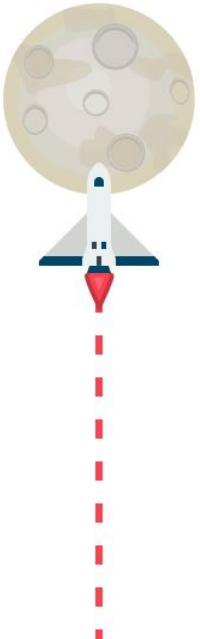
Let's print it out a little better:

```
books = []
book = input('What book are you returning? ')
while book:
    books.append(book)
    book = input('What book are you returning? ')
books.sort()
for book in books:
    print(book)
```

Test it out!

Try the fourth question now!

GROK LEARNING



Any Questions?

Find me at:
@groklearning
nicky@groklearning.com

GROK LEARNING