print("Hello, World!")

NCSS Challenge - Beginners
Week 1 Part 1
What is programming?

It’s less like trying to decipher the matrix, and more like following a recipe.
Where do we start?

- Writing your first program;
- Python strings and variables;
- Reading user input;
- Reusing variables.
What does this cover?

- Define simple problems, and describe and follow a sequence of steps and decisions (algorithms) needed to solve them (ACTDIP010).
- Implement simple digital solutions as visual programs with algorithms involving branching (decisions) and user input (ACTDIP011).
Your first program

Hello, World!
How do we start

print('Hello, World!')
Pedagogical Philosophy - Get your hands dirty!

Interactive code snippets let students explore:

```python
print('Hello, World!')
```

You can edit and run any example in Grok by clicking the button. Try changing 'Hello, World!' to 'Hi!', and running it again.
Pedagogical Philosophy - Get your hands dirty!

- Encourage students to experiment:

Here we accidentally put `write` instead of `print`:

```python
write('Hi There')
```

Python doesn't know they mean the same thing, so gives an error:

```
Traceback (most recent call last):
  File "program.py", line 1, in <module>
    write('Hi There')
NameError: name 'write' is not defined
```

Python displays the error (it does not recognise the name `write`) in red, including the type (`NameError`) and where it occurred (`line 1`).
Test it out!
Try the first question now!
Pedagogical Philosophy - Precision is important!

- Auto-marking guides students to a correct solution:

Number #3: Failed (3 tests passed)

- Testing that the words are correct.
- Testing that the whitespace is correct.
- Testing that the punctuation is correct.

*Your program did not use the correct capitalisation.*

Your program output:

```
hello, world!
```

When it was meant to output:

```
Hello, World!
```

It should print *exactly* what the question asks for.
Computers are very, very dumb. Very fast, but very dumb. Computers are so dumb that they only see differences, not similarities.
Pedagogical Philosophy - Formative assessment

- Each question tests a concept just introduced
- Notes introduce a concept
- Students explore it through interactive notes
- Students apply it in answering the question
- Students get immediate feedback on whether they have understood the concept.
2 Strings and Variables
Strings of Characters

print('abc ABC 123 @!?.#')

- String is short for *string of characters*
- Can contain letters, digits, punctuation & spaces
- Can be bound by single or double quotes:
  - print('She said, "hello!" to us.')
  - print("Weren't you listening?")
Teacher Aside!

- Using either single quotes or double quotes is fine.
- You can also delimit strings with triple quotes:

  ```python
  message = """This is a message containing punctuation: ', " and other things!"""
  ```
Joining Messages Together

○ We can join two strings together by using addition, called *concatenation*
○ Can contain letters, digits, punctuation & spaces
○ Can be bound by single or double quotes:
  ● `print('Harry' + 'Potter')`
    → HarryPotter
  ● `print('Harry' + ' ' + 'Potter')`
    → Harry Potter
Saving Strings in Variables

- A variable lets you store a value for later use
- Variables have names which we use both to set and to get its value:

```python
name = 'Hermione'
print('I saw ' + name + '.

→ I saw Hermione.'
```
Changing Strings in Variables

- The contents of variables can be changed, much like writing over a file:

```python
name = 'Hermione'
print('I saw ' + name + ' .')
name = 'Ron'
print('I saw ' + name + ' .')

→ I saw Hermione.
→ I saw Ron.
```
Test it out!

Try the second question now!
Input from the User
You can ask the user for information that you can save directly in a variable:

```python
name = input('What is your name? ')  
print(name)
```

Whatever the user types in is saved in the variable name, and then printed out!
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.
Test it out!

Try the third question now!
Any Questions?

Find me at:
@groklearning
nicky@groklearning.com