Monty Python (aka The Pythons) were a British surreal comedy group who created the sketch comedy show *Monty Python's Flying Circus*, that first aired on the **BBC** on October 5, 1969. Loosely structured as a sketch show, but with an innovative **stream-of-consciousness** approach, it pushed the boundaries of what was acceptable in style and content.
① Project tools
- Tasks. Completed.
- Files supported by PyCharm Edu
- Sandbox: create files and folders

② Course Progress

③ Console
- Clear output

④ Code Editor

⑤ Task Description
- Next task
- Previous task
- Start task again
- View hints
Introduction to Python

- Python - an open-source general-purpose interpreted programming language
- Programming constructs (if else, while, for)
- Built-in functions - upper(), random(), sorted()...
- **Type in** Editor (command or expression)
  ```python
  print("Hello, world! My name is Susan")
  ```
- **Run** – Output (Python Console)

```python
print("Hello, world! My name is Susan")
```
What is a **Variable**?

- **Variable** stores a **value**
- **Assignment statement** puts a value into a variable
  - General form: \[ \text{variable} = \text{expression} \]

  \[
  \begin{align*}
  \text{age} & = 20 + 1 \\
  \end{align*}
  \]

  - **Assigns result of expression, 21, to variable called age**
Modifying Variables

- **Initialize** a variable (assign first value)
  \[ \text{day} = 2 \]
- Change a variable (add 3 to it)
  \[ \text{day} = \text{day} + 3 \]
- **Increment**
  \[ \text{day} = \text{day} + 1 \]
- **Decrement**
  \[ \text{day} = \text{day} - 2 \]

What value does \text{day} have now?
Variable Names

- **Rules**
  - Must begin with a letter
  - May be followed by letters, digits, underscore (_)

- **Case-sensitive**
  - lower_case (variable names)
  - UPPER_CASE (constants)

- **Mnemonic** (makes sense!)

- **Python internal use:** begin/end with 2 underscores

- **AVOID:** 'I' (letter el), 'O' (letter OH), 'I' (letter EYE) as single character variable names

```plaintext
what      easy_peasy      2be_or_not      0
```
# Arithmetic Operators

<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sum</strong></td>
<td>$1 + 2$</td>
</tr>
<tr>
<td><strong>difference</strong></td>
<td>$10 - \text{sum}$</td>
</tr>
<tr>
<td><strong>result</strong></td>
<td>$7 / 2$</td>
</tr>
<tr>
<td><strong>product</strong></td>
<td>$2 \times -1.23$</td>
</tr>
<tr>
<td><strong>remainder</strong></td>
<td>$7 % 4$</td>
</tr>
<tr>
<td><strong>exponent</strong></td>
<td>$2 \times 3$</td>
</tr>
</tbody>
</table>

What are the *values* of these variables?
Data Types

day    = 23               # int
pi     = 3.14             # float
name   = "katie"          # string ‘Jon’
isFun  = True            # boolean: False
quiz   = [8, 7, 9]        # list
labs   = ["html", 5]      # list
more Variables

hw1 = hw2 = 3.3

hw3, hw4, hw5 = 2, 6, 4

hw6 = hw1

hw1 = 1

hw3 = hw1 + hw6

What is the value of hw3?
b1 = '2b'

b2 = b1 + " OR "

b1 = 2

print( b2, end="" )  # suppress newline
print( b1 )

What gets printed?
Type in Editor

# Prompt for food
food = input("Enter favorite food: ")

# Display
print("Let's have ", food, " tonight!")

Run - Output

Enter favorite food: pizza
Let's have pizza tonight!
Operator Order and Precedence

1 + 2 − 3 + 4

1 + 2 * 3 − 4

<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>* / %</td>
<td>Multiply, divide, modulo</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition and subtraction</td>
</tr>
<tr>
<td>= %= /= -= += *=</td>
<td>Assignment operators</td>
</tr>
</tbody>
</table>
# Table of Precedence

<table>
<thead>
<tr>
<th>OPERATOR</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>**</td>
<td>Exponentiation (raise to the power)</td>
</tr>
<tr>
<td>* / %</td>
<td>Multiply, divide, modulo</td>
</tr>
<tr>
<td>+ -</td>
<td>Addition and subtraction</td>
</tr>
<tr>
<td>&lt;= &lt; &gt; &gt;=</td>
<td>Comparison operators</td>
</tr>
<tr>
<td>&lt; &gt; == !=</td>
<td>Equality operators</td>
</tr>
<tr>
<td>= %= /= -= += *=</td>
<td>Assignment operators</td>
</tr>
<tr>
<td>not or and</td>
<td>Logical operators</td>
</tr>
</tbody>
</table>