Functions and Variables

Click `Labs` > `launch` button

Jupyter notebook on Ed Lesson

- code cell
- text cell (markdown syntax)
- run (▶️ or `shift+enter` or `ctrl+enter`)
- autocomplete / syntax highlighting

Guide to Using Your Lab Notebook

- Structure
- During class
- Playgrounds
- For later

1. Hello World

```
Hello, World!
```

Anatomy

```
print("Hello, World")
```

- Function: `print()`
- Argument: `"Hello, World"`
- Side effect: print to the screen

Bugs

```
print("Hello world"
```

```
  Cell In[1], line 1
    print("hello world"
                       ^
SyntaxError: incomplete input
```

2. Hello to You

```
What's your name? John
Hello, John!
```

Anatomy

```
answer = input("What's your name? ")
```

- Function: `input()`
- Argument: `"What's your name? "`
- Side effect: prompt the user and wait for input
- Return values: user input
- Variable: `answer`

Hello to You?

```
answer = input("What's your name? ")
print("Hello, answer")
```

Joining strings and variables (+)

```
answer = input("What's your name? ")

print("Hello " + answer)
```

Joining strings and variables (multiple arguments)

```
answer = input("What's your name? ")

print("Hello", answer)
```

```
help(print)
```

```
Help on built-in function print in module builtins:

print(*args, sep=' ', end='\n', file=None, flush=False)
    Prints the values to a stream, or to sys.stdout by default.

    sep
      string inserted between values, default a space.
    end
      string appended after the last value, default a newline.
    file
      a file-like object (stream); defaults to the current sys.stdout.
    flush
      whether to forcibly flush the stream.
```

Or, you can refer to the documentation online:

https://docs.python.org/3/library/functions.html#print

Joining strings and variables (f-string)

```
answer = input("What's your name? ")

# print("Hello, answer")
print(f"Hello, {answer}")
```

Comments

```
# + operator
print("Hello " + answer)

# multiple arguments
print("Hello", answer)

# f-string
print(f"Hello {answer}")
```

3. Personalized Introduction 💻

Requirements:

- Use Python's input() function to prompt the user for their name and age.
- Store these values in respective variables.
- Utilize the print() function and string formatting to display a message that says "Hello, my name is xx. I am xx years old." where the xx's are replaced with the user's name and age.

Expected Outputs:

```
What's your name? Emily
How old are you? 25
Hello, my name is Emily. I am 25 years old.
```

4. Uncooperative user

```
What is your name?     john
Hello, John

What is your name? jAnE doE
Hello, Jane Doe
```

String method

https://docs.python.org/3/library/stdtypes.html#string-methods

strip()

```
answer = input("What's your name? ")

answer = answer.strip()

print("Hello " + answer)
```

capitalize()

```
answer = input("What's your name? ")

answer = answer.strip().capitalize()

print("Hello " + answer)
```

`title()`

```
answer = input("What's your name? ")

answer = answer.strip().title()

print("Hello " + answer)
```

replace()

```
sentence = "I like apples, but I don't like green apples."
new_sentence = sentence.replace("apples", "oranges")
print(new_sentence)
```

split()

```
sentence = "I like apples, but I don't like green apples."
words = sentence.split()
print(words)
```

5. Hello Function

```
hello()
# Output: Hello, World!

hello("John")
# Output: Hello, John
```

def

```
def hello():
    print("Hello world")
answer = input("What's your name? ")
hello()
```

Arguments

```
def hello(to):
    print("Hello ", to)
answer = input("What's your name? ")
hello(answer)
```

Arguments with default values

```
def hello(to="world"):

    print("Hello ", to)

answer = input("What's your name? ")

hello(answer)
# Output: Hello name

hello()
# Output: Hello world
```

main

```
# Write main first
def main():

    answer = input("What's your name? ")

    hello()

# Then write hello
def hello():
    ...

main()
```

```
# Write main first
def main():

    answer = input("What's your name? ")

    hello()

# Then write hello
def hello():

    print("Hello ", answer)

main()
```

Scope

```
def main():
    answer = input("What's your name? ")
    hello(answer)

def hello(to="world"):
    print("Hello ", to)

main()
```

return

```
def main():
    answer = input("What's your name? ")
    message = hello_message(answer)
    print(message)

def hello_message(to="world"):
    return "Hello " + to

main()
```

6. Personalized Introduction 2 💻

Requirements:

- Define a function `ask_name()` that prompts the user for their name using Python's `input()` function and returns the name.
- Define another function `ask_age()` that prompts the user for their age and returns the age.
- Define a function `introduce_message()` that takes name and age as parameters and returns a string in the format "Hello, my name is [name]. I am [age] years old."

Expected Outputs:

```
Name: Emily
Age: 25
Hello, my name is Emily. I am 25 years old.
```

7. calculator

```
Enter a number: 5
Enter another number: 3
8
```

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")
    print(x + y)

calculator()
```

integer and `int()`

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")
    z = int(x) + int(y)
    print(z)

calculator()
```

`type()` to check variable type

```
x = input("Enter a number: ")
print(type(x))

x = int(x)
print(type(x))
```

style

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")
    z = int(x) + int(y)
    print(z)
```

vs.

```
def calculator():
    x = int(input("Enter a number: "))
    y = int(input("Enter another number: "))
    print(x+y)
```

vs.

```
def calculator():
    print(int(input("Enter a number: ")) + int(input("Enter another number: ")))
```

floating-point numbers and `float()`

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")

    z = float(x) + float(y)
    print(z)

calculator(1+900)
```

type conversion functions

- `int()`
- `float()`
- `str()`

...

float formatting

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")

    z = float(x) + float(y)
    print(f"{z:,}")

# try 1 and 999
calculator()

# Output: 1,000
```

float formatting

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")

    z = float(x) / float(y)
    print(f"{z:.2f}")

# try 2 and 3
calculator()

# Output: 0.67
```

round()

```
def calculator():
    x = input("Enter a number: ")
    y = input("Enter another number: ")

    z = round(float(x) / float(y), 2)
    print(z)

# try 2 and 3
calculator()

# Output: 0.67
```

8. Personalized Introduction 3 💻

Requirements:

- Define a function `ask_birthyear()` that prompts the user for their birth year and returns it.
- Define another function `calc_age()` that takes the birth year as a parameter and returns the calculated age based on the current year (2023).
- Utilize the previously defined `ask_name()` and `introduce_message()` functions.
- Define a `main()` function that orchestrates the execution of these functions and prints the final introduction message.

Expected Outputs:

```
What's your name? Emily
What's your birth year? 1998
My name is Emily and I am 25 years old.
```