labsheet-01

July 19, 2023

1 Exercise 2

1.1 Basic Python Programming Language

[1]:

```
# Variables
my_variable = 10
print(my_variable)
```

```
10
```

[2]:

```
# Data types
integer = 10
float = 10.0
string = "Hello, world!"
list = [1, 2, 3]
dictionary = {"key1": "value1", "key2": "value2"}

print(integer, type(integer))
print(float, type(float))
print(string, type(string))
print(list, type(list))
print(dictionary, type(dictionary))
```

```
10 <class 'int'>
10.0 <class 'float'>
Hello, world! <class 'str'>
[1, 2, 3] <class 'list'>
{'key1': 'value1', 'key2': 'value2'} <class 'dict'>
```

[3]:

```
# Operators
print(10 + 20)
print(10 - 20)
print(10 * 20)
print(10 / 20)
print(10 % 20)
```

```
30
-10
200
```

```
0.5
10
```

[4]:

```
# Control flow
if 10 > 20:
    print("10 is greater than 20")
else:
    print("10 is not greater than 20")
```

```
10 is not greater than 20
```

[5]:

```
for i in range(10):
    print(i)
```

```
0
1
2
3
4
5
6
7
8
9
```

[6]:

```
while i < 10:
    print(i)
    i += 1
```

```
9
```

[7]:

```
# Function
def greet(name):
    print(f"Hello, {name}!")

greet("Alice")
```

```
Hello, Alice!
```

[ ]:

```
def multiply_numbers(num1, num2=1):
    return num1 * num2

result = multiply_numbers(5)
print(result)  # Output: 5
```

2 Exercise 3

2.1 Importing the python control and matplotlib modules

[9]:

```
import control
import matplotlib
```

[10]:

```
print(control.__version__)
print(matplotlib.__version__)
```

```
0.9.4
3.7.1
```

3 Exercise 3

Basic Jupyter Notebook

[11]:

```
# Markdown
```

Markdown supports tables:

Column 1	Column 2
Row 1, Column 1	Row 1, Column 2
Row 2, Column 1	Row 2, Column 2

You can display code blocks with syntax highlighting:

```
def multiply_numbers(num1, num2=1):
    return num1 * num2

result = multiply_numbers(5)
print(result)  # Output: 5
```

You can include mathematical equations using LaTeX syntax:

$$m_n = k_p * e_n + \frac{k_e * T}{T_{reset}} \sum_{i=0}^{n} e_i + k_d \frac{e_n - e_{n-1}}{\delta t} + m_R$$

You can use basic formatting options like *italic*, **bold**, and `code` highlighting.

Markdown supports both ordered and unordered lists:

- Unordered List Item 1
- Unordered List Item 2
- Unordered List Item 3

1. Ordered List Item 1
2. Ordered List Item 2
3. Ordered List Item 3

You can add - Google - `labsheet-01.ipynb`

Embedding images is also possible:

[12]:
```
# Widgets
```

[13]:
```
import ipywidgets as widgets
from IPython.display import display

# Create a text input widget
text_input = widgets.Text(value='Hello', description='Enter text:')

# Create a button widget
button = widgets.Button(description='Click me!')

# Create a dropdown menu widget
dropdown = widgets.Dropdown(options=['Option 1', 'Option 2', 'Option 3'],␣
 ↪description='Select an option:')

# Create a slider widget
slider = widgets.IntSlider(value=50, min=0, max=100, step=1,␣
 ↪description='Slider:')

# Create an output widget
output = widgets.Output()

# Event handler for the button click
def on_button_click(button):
    with output:
        print(f"Text: {text_input.value}")
        print(f"Selected option: {dropdown.value}")
        print(f"Slider value: {slider.value}")

# Assign the event handler to the button's on_click event
button.on_click(on_button_click)

# Display the widgets
display(text_input)
display(button)
display(dropdown)
display(slider)
display(output)
```

```
Text(value='Hello', description='Enter text:')
```

```
Button(description='Click me!', style=ButtonStyle())
```

```
Dropdown(description='Select an option:', options=('Option 1', 'Option 2', 'Option 3'), value='Option 1')
```

```
IntSlider(value=50, description='Slider:')
```

```
Output()
```

[ ]:

```

```