if - elif - else

Week 3

`if..else` statements

```
if <condition>:
    <block of code 1>
else:
    <block of code 2>
```

- If `condition` is `True`, we execute `<block of code 1>`
- If `condition` is `False`, we execute `<block of code 2>`

Example

```
age = 18

if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote.")
```

File: `Example8.py`

if..elif..else

- `elif` is shorthand for **else if**

```
if <condition>:
    <statement(s)>
elif <condition>:
    <statement(s)>
elif <condition>:
    <statement(s)>
          ⋮
else:
    <statement(s)>
```

`if..elif..else`

- This is an extension of the basic `if..else` statement, allowing for multiple conditions to be checked in sequence.
- This is useful when you have more than two possible outcomes.
- `if` checks the first condition. If it is true, the corresponding block of code runs, and the `elif` and `else` blocks are skipped.
- `elif` means "`else if`". This allows you to check additional conditions if the previous `if` or `elif` conditions were false. You can have multiple elif blocks.
- `else` part is executed if none of the preceding `if` or `elif` conditions are true.

Example

```
if score >= 90:
    print("Grade: 5")
elif score >= 80:
    print("Grade: 4")
elif score >= 70:
    print("Grade: 3")
elif score >= 60:
    print("Grade: 2")
elif score >= 50:
    print("Grade: 1")
else:
    print("Grade: 0")
```

`if..elif..else`

- **Only one** block among the several `if..elif..else` blocks is executed!
- Every `if` condition can have *at most one* `else` block
- `else` part can be also missing.
- There can several `elif` blocks following each other.

File: `Example9.py`

File: `Example10.py`