Run-time errors

Week 9

TypeError

Raised when an operation or function is applied to an object of **inappropriate** type.

```
x = "hello" + 5
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    x = "hello" + 5
TypeError: can only concatenate str (not "int") to str
```

NameError

Raised when a name (e.g. variable or function) is **not found**.

```
print(y)
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    print(y)
NameError: name 'y' is not defined
```

IndexError

Raised when an **index** is out of **range**.

```
my_list = [1, 2, 3]
print(my_list[4])
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    print(my_list[4])
IndexError: list index out of range
```

AttributeError

You try to call a method or attribute that does not exist on the class.

```
my_list = [1, 2, 3]
my_list.append(4)  # This works
my_list.add(5)     # This will raise an AttributeError
Traceback (most recent call last):
  File "<pyshell#12>", line 1, in <module>
    my_list.add(5)     # This will raise an AttributeError
AttributeError: 'list' object has no attribute 'add'
```

ValueError

When a function receives an argument of the right type but inappropriate value.

```
int("5")    # this works
5
int("five") # this raises ValueError
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    int("five") # this raises ValueError
ValueError: invalid literal for int() with base 10: 'five'
```

ZeroDivisionError

Raised when division or modulo by zero takes place:

```
10 / 0
Traceback (most recent call last):
  File "<pyshell#6>", line 1, in <module>
    10 / 0
ZeroDivisionError: division by zero
```

FileNotFoundError

Raised when a file or directory is requested but doesn't exist.

```
open("non_existent_file.txt", "r")
Traceback (most recent call last):
  File "<pyshell#8>", line 1, in <module>
    open("non_existent_file.txt", "r")
FileNotFoundError: [Errno 2] No such file or directory: 'non_existent_file.txt'
```

`Errno 2` is a standard error code in operating systems: https://learn.microsoft.com/en-us/windows/win32/debug/system-error-codes--0-499-

ImportError

Occurs when the interpreter cannot find the module you're trying to import. This usually happens if the module doesn't exist, is misspelled, or is not installed.

```
import non_existent_module
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    import non_existent_module
ModuleNotFoundError: No module named 'non_existent_module'
```

Understand error messages

test.py - C:/Users/Jouni/OneDrive - LUT University/Desktop/test.py (3.11.9)

File Edit Format Run Options Window Help

```
def divide(x,y):
    return x / y

print(divide(7,0))
```

```
Traceback (most recent call last):
  File "C:/Users/Jouni/OneDrive - LUT University/Desktop/test.py", line 4, in <module>
    print(divide(7,0))
  File "C:/Users/Jouni/OneDrive - LUT University/Desktop/test.py", line 2, in divide
    return x / y
ZeroDivisionError: division by zero
```

Understand error messages

`Traceback` indicates the path of function calls that led to the error:

- `line 4, in <module>:` The error originated from **line 4** in the within the main program block (<module>).
- `print(divide(7,0)):` This line caused the error. It calls the `divide` function with parameters 7 and 0.
- `line 2, in divide:` The error occurred during the execution of the divide function on **line 2**.
- `return x / y`: This is the line in the divide function where the division by zero happens.
- `ZeroDivisionError: division by zero:` This is the actual error message, indicating that you attempted to divide by zero, which is not allowed in Python.