



SKILLPILLS

Skill Pill: Introduction to Programming

Extra: Error Messages

Vsevolod Nikulin

vsevolod.nikulin@oist.jp

August 12th, 2020



What will happen if I run this?

```
def half_list(input_list):  
    length = len(input_list)  
    halved_list = l[length/2 : ]  
    return halved_list  
  
def print_half(input_list):  
    half = half_list(input_list)  
    print(half)  
  
my_list = [1,2,3,4,5]  
print_half(my_list)
```

Traceback (most recent call last):

```
File "...", line ..., in <module>  
    print_half(my_list)
```

```
File "...", line ..., in print_half  
    half = half_list(input_list)
```

```
File "...", line ..., in half_list  
    halved_list = input_list[length/2 : ]
```

TypeError: slice indices must be integers or None ...

This output contains information about the location of occurred error. Nested function calls are listed from top to bottom.

What will be the result of this?

```
(-2**2)**0.5
```

`**` is the exponentiation function.

What will be the result of this?

```
(-2**2)**0.5 == 2j
```

j is the imaginary unit.

Can you see why is that?

Sometimes error is not manifested in an error message but rather in unexpected result.

```
def safe_sqrt(x):  
    assert(x >= 0.0)  
    return x**0.5
```

Assert function takes Boolean value and raises an error when it is False. It acts as a sanity check. Do not think "this will never happen, why do I need this?". It may and it will happen at some point.