INFO5002: Intro to Python for Info Sys

Advanced Functions

PCC 133-135

Northeastern University

Slides created by: Zachary Doucet

Additional functional features

- You can define a function's argument(s) as optional by providing a default value.

```
def increment(input, by=1):
    return input + by
```

WARNING: When using default values make sure that **all non default values appear before** in function's signature.

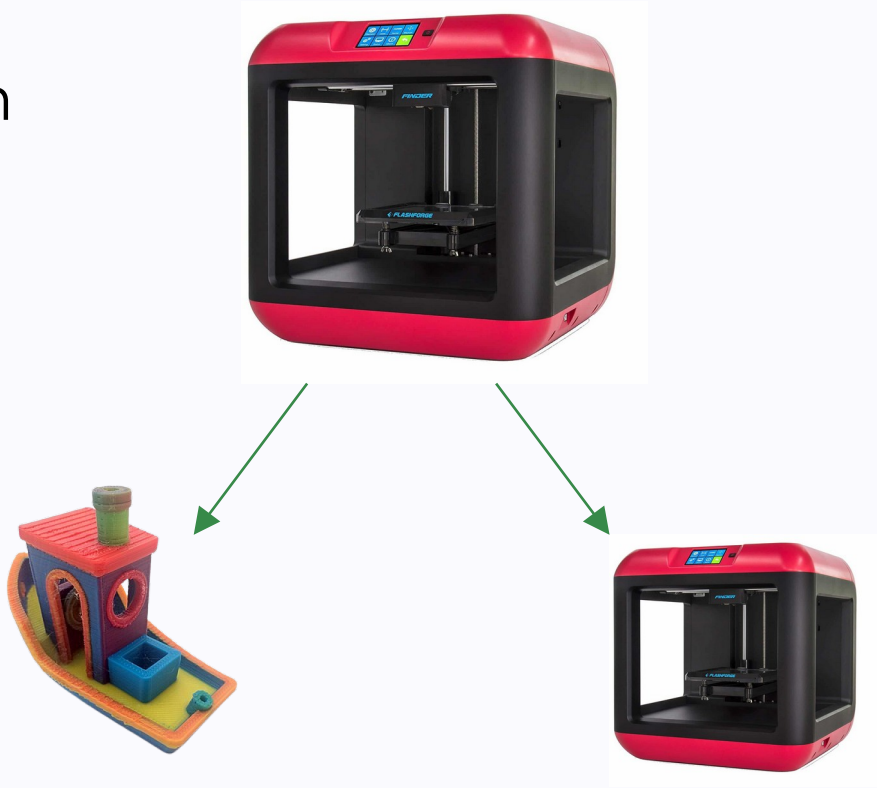
- If we want to not rely on argument positions when calling a function we can use keyword arguments.

```
def calculate_mortgage_payment(principal, downpayment, interest,
        is_fixed, term, amortisation):
    ...
```

```
calculate_mortgage_payment(principal=1_000_000, interest=0.052,
   downpayment=45_000, term=3, amortisation=35, is_fixed=True)
```

Higher Order Functions

- Functions usually return data.
- What if we return a function?



Create and return a function

```
def create_greeting(person_name):
    def tell_person(message):
        print("Hey " + person_name + ". " + message)
    return tell_person
```

```
x = create_greeting("Bobbie")
x("Want to join tomorrow?")
# Hey Bobbie. Want to join tomorrow?
x("Don't forget class Friday!")
# Hey Bobbie. Don't forget class Friday!
```

Recursion

What happens if I call myself?

```
def countdown(t):
  print(t)
  countdown(t-1)
```

```
countdown(10)
```

```
10
9
...
-980
```

```
File "<python-input-2>", line 3, in countdown
  countdown(t-1)
  ~~~~~~~~~~^^^^
[Previous line repeated 988 more times]
RecursionError: maximum recursion depth exceeded
```

Source: Sergey Pykhonin