

Avanceret Programming (Uge 36)

Christian Gram Kalhauge (CKL)

Section 1

Today

Last Time

- Variables
- Strings, List, Tuples
- IO: input, print
- If Statements
- While and For-each

This Day

- ① Today
- ② Missing Pieces
- ③ Functions
- ④ Objects

Section 2

Missing Pieces

Sets

```
>>> set()  
set()
```

```
>>> set([10, 4, 2, 10])  
{10, 2, 4}
```

```
>>> {10, 15}  
{10, 15}
```

Dictionaries

```
>>> dict()  
{}
```

```
>>> dict(x = 10)  
{'x': 10}
```

```
>>> {'x': 10, 'x': 15}  
{'x': 15}
```

Dictionaries (Iter)

```
>>> for k in {'x': 10, 'y': 15}:
...     print(k)
x
y
```

```
>>> for item in {'x': 10, 'y': 15}.items():
...     print(item)
('x', 10)
('y', 15)
```

```
>>> for k, v in {'x': 10, 'y': 15}.items():
...     print(k, v)
x 10
y 15
```


Comprehensions (link)

```
>>> [i for i in range(4) if i % 2 == 0]  
[0, 2]
```

```
>>> {i % 2 for i in range(4)}  
{0, 1}
```

```
>>> {i: i % 2 for i in range(4)}  
{0: 0, 1: 1, 2: 0, 3: 1}
```

str vs repr

```
>>> str('hello')
'hello'
>>> repr('hello')
"'hello'"

>>> f"- {'hello'!r} -"
"- 'hello' -"
>>> f"- {'hello'!s} -"
'- hello -'
```

Files - input

```
>>> with open("example.txt") as f:
...     print(f"first line = {f.readline()!r}")
...     for line in f:
...         print(repr(line))
first line = 'This is line 1\n'
'This is line 2\n'
'and so on.\n'
```

Files - output

```
.>> with open("other.txt", "w") as f:  
...     f.write("Hello\n")  
...     print("Hello!", file=f)
```

```
>>> import sys  
>>> x = sys.stdout.write("Hello!\n")  
Hello!  
>>> x  
7
```

Assignment 1 (CSV):

```
>>> from csv import DictReader
>>> with open("vgsales.csv") as f:
...     reader = DictReader(f)
...     for line in reader:
...         break
```

Assignment 1, part 2

```
>>> for k, v in line.items():  
...     print(f"{k:>14}: {v!r}")  
          Rank: '1'  
          Name: 'Wii Sports'  
    Platform: 'Wii'  
          Year: '2006'  
          Genre: 'Sports'  
    Publisher: 'Nintendo'  
    NA_Sales: '41.49'  
    EU_Sales: '29.02'  
    JP_Sales: '3.77'  
    Other_Sales: '8.46'  
    Global_Sales: '82.74'
```

Assignment 1

- 1 Find the sum of all global sales (`Global_Sales`)
- 2 Find all publishers
- 3 Find the highest grossing publisher
- 4 Do it by year

Section 3

Functions

A Simple Function

```
>>> def say_happy(what):  
...     print(f"Don't worry, be {what}!")  
>>> say_happy("happy")  
Don't worry, be happy!
```

return to basics

```
>>> def give_me_a_number():  
...     return 42  
>>> give_me_a_number()  
42
```

Keyword arguments

```
>>> def caesar(msg, offset=3):  
...     return ''.join(chr(ord(c) + offset) for c in msg)  
  
>>> caesar("Secret message")  
'Vhfuhw#phvvdjh'  
  
>>> caesar("Vhfuhw#phvvdjh", offset=-3)  
'Secret message'
```

Arbitrary Arguments (*args)

```
>>> def print_everything(*args):  
...     for count, thing in enumerate(args):  
...         print(f'{count}. {thing}')  
...  
>>> print_everything('apple', 'banana', 'cabbage')  
0. apple  
1. banana  
2. cabbage
```

Arbitrary Arguments (**kwargs)

```
>>> def table_things(**kwargs):  
...     for name, value in kwargs.items():  
...         print(f'{name} = {value}')
```

...

```
>>> table_things(apple = 'fruit', cabbage = 'vegetable')
```

apple = fruit
cabbage = vegetable

Many args

```
>>> def function(a, b, c):  
...     return a + b + c  
>>> numbers = [1, 2, 3]  
>>> function(*numbers)  
6  
>>> function(*[1,2,3])  
6  
>>> byname = { "a": 1, "b": 2, "c": 3 }  
>>> function(**byname)  
6
```

Anonumus functions (Lambdas)

```
>>> fn = lambda x : x
>>> fn(10)
10
>>> fn = lambda x, y, z, bang = "!" : x + y + z + bang
>>> fn("Hel", "lo, ", "World")
'Hello, World!'
>>> fn = lambda *args: args
>>> fn('a', 'b', 'c')
('a', 'b', 'c')
```

Assignment 1 1/2

- Fix assingment 2 from yesterday where you use a dict of lambdas.
- Use the operator library

```
from operator import add, sub, floordiv, mult
```


Scope

```
>>> var = 1
>>> def globalfn():
...     return var
>>> globalfn()
1
>>> def localfn():
...     var = 2
...     return var
>>> localfn()
2
>>> var
1
```

Scope (cont.)

```
>>> var = 1
>>> def globalfn2():
...     global var
...     var = 2
...     return var
>>> globalfn2()
2
>>> var
2
```

Closures

```
>>> def counter():
...     counts = 0
...     def inner():
...         nonlocal counts
...         counts += 1
...         return counts
...     return inner
>>> count = counter()
>>> count()
1
>>> count()
2
```

Assignment 2

Build a logger creator:

```
from datetime import datetime
```

```
def logger(file):  
    def inner(msg, level="INFO"):  
        ...
```

```
log = logger(sys.stdout);  
log("Hello, World!")  
log("Second log", level="DEBUG")
```

```
17:30 - INFO - Hello, World!
```

```
17:31 - DEBUG - Hello, World!
```