Functions

Tomas Svoboda
http://cmp.felk.cvut.cz/~svoboda
Programming Essentials, EECS, CTU in Prague

```python
__author__ = 'svoboda'

def compute_monthly_payments(P,N,r):
    c_multiplicator = 1
    for i in range(1,N):
        c_multiplicator = c_multiplicator + (1+r)**i
    return (((1+r)**N)*P) / c_multiplicator

def get_amount_owed(P,r,c,m):
    if m==0:
        return P
    previous_amount = get_amount_owed(P,r,c,m-1)
    return (1+r)*previous_amount - c

P,R,Y = 12000, 12, 1
N = 12*Y
r = (R/12)/100
print("My input:",P,R,Y,r)
c = compute_monthly_payments(P,N,r)
print("My monthly playments will be: ", c)
# simple check
diff = N*c - P
print('Difference: ',diff)
# better check
end_amount = get_amount_owed(P,r,c,N)
print("end amount", end_amount, abs(end_amount)<1e-9)
```

why functions?

- re-using code
- enforcing logical structure into the code
- easier debugging
- code readability

built-in functions

- https://docs.python.org/3.4/library/functions.html
- `abs(x), len(s), min(iterable), …`

libraries, math, …

- https://docs.python.org/3.4/library/math.html

```
1 import math
2
3 a = math.sqrt(9)
4 radius = 3
5 area = math.pi*radius**2
```

basic structure

```
1 def function_name(input_argument):
2     '''function_name computes ...'''
3     # some code here
4     return variable_to_be_returned
```

What is a good function name?

call/invoke a function

```
1 def function_name(input_argument):
2     '''function_name computes ...'''
3     # some code here
4     return variable_to_be_returned
5
6 if __name__ == "__main__":
7     a = function_name(34)
```

a simple function: area

```
1 def area(a,b):
2       s = a*b
3       return s
```

What can be done better? What is missing?

doc string, comments

- think about re-using of the function
- what needs to be known

```
1 def rectangle_area(a,b):
2     '''
3     computes area of a rectangle
4     Inputs:
5     a,b rectangle sides, numerical values expected
6     Returns:
7     numerical value
8     '''
9     s = a*b
10    return s
```

scope of variables

- visualization of the area function call

distance function

- how to make it more flexible
- L1 (manhattan, taxicab), L2 (Euclidean) most common
- *… live-coding session*

default argument

- `def distance(x1,y1,x2,y2,p=1):`

- when called/invoked with 4 arguments p=1

keyword arguments

- `kwarg=value` when calling the function
- makes calling function more flexible …
- … and sometimes more readable
- use cautiously

boolean functions

- return `True` or `False`

- `is_divisible(x)`

- `has_children(p)`

- `isinstance(x,type) # built-in`

readability

- some conditionals may be complex
- or difficult to understand without comments

```
# if adult and has some children
if age>18 and sql(SELECT Parent FROM …):
    tax = 0.1
```

```
if is_adult(a) and has_children(a):
    tax = 0.1
```

program flow

- imports
- functions definitions
- main program
- not that strict, actually. But …