Python workshop

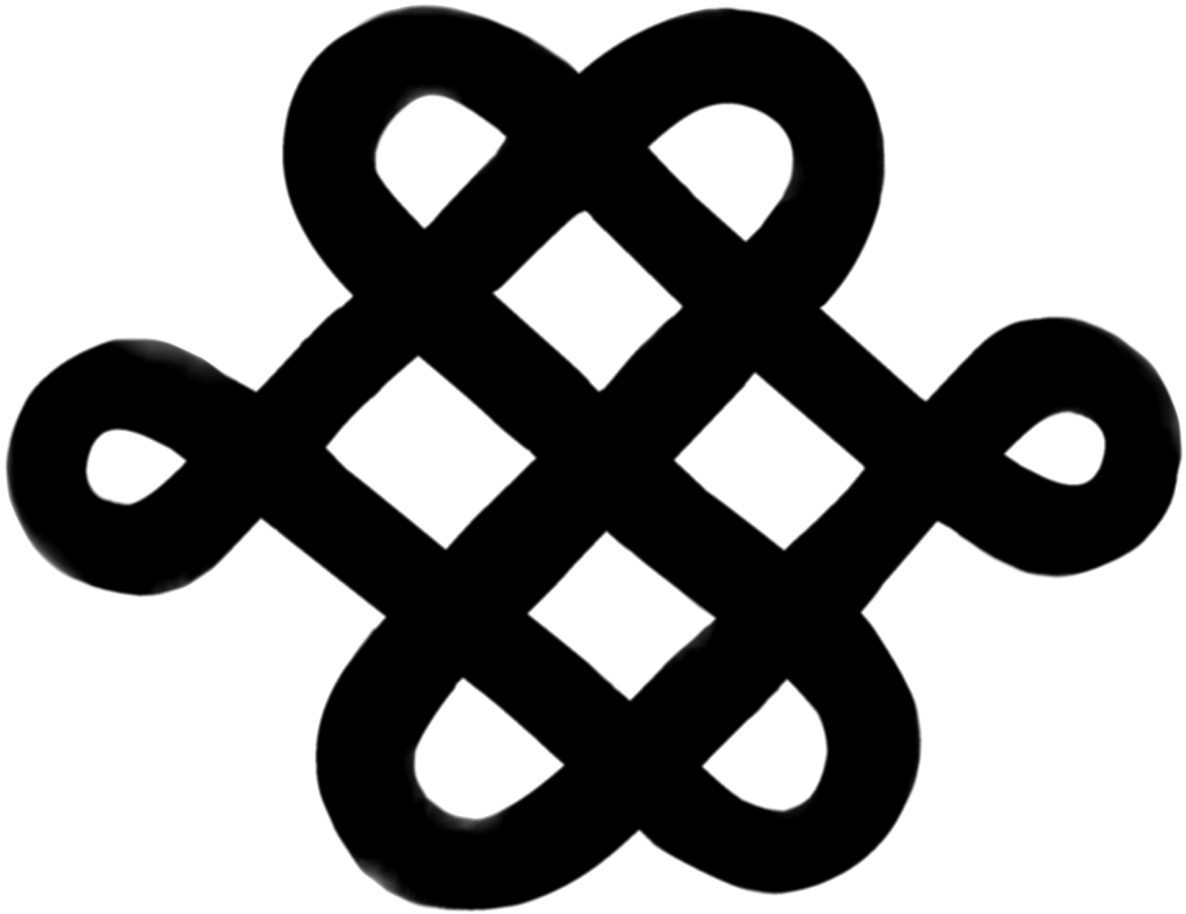
Week 3: Loops and strings

barbera@van-schaik.org

Overview of this workshop series

- Week 1: Writing your first program
- Week 2: Make choices and reuse code
- Week 3: Loops and strings
- Week 4: Files and lists
- Week 5: Dictionaries and tuples

Acknowledgments: Structure of the workshop follows the book
“Python for informatics” by Charles Severance.
Several examples are from this book or the accompanying slides.



Loops

99 bottles of beer

99 bottles of beer on the wall, 99 bottles of beer.

Take one down, pass it around, 98 bottles of beer on the wall...

98 bottles of beer on the wall, 98 bottles of beer.

Take one down, pass it around, 97 bottles of beer on the wall...

.

.

No more bottles of beer on the wall, no more bottles of beer.

Go to the store and buy some more, 99 bottles of beer on the wall...

Try this script and fix the error:

bottles-of-beer-contains-error.py

You can refine this script to deal with the case of 1 **bottle** of beer

99 bottles of beer


```
n = 99

while n>0:

    if n > 0:

        print n, "bottles of beer on the wall,", n ,"bottles of beer."

        n = n - 1

        print "Take one down, pass it around,", n, "bottles of beer on the wall...\n"



    if n == 0:

        print "No more bottles of beer on the wall, no more bottles of beer."

        print "Go to the store and buy some more, 99 bottles of beer on the wall...\n"
```

Declaration of a variable

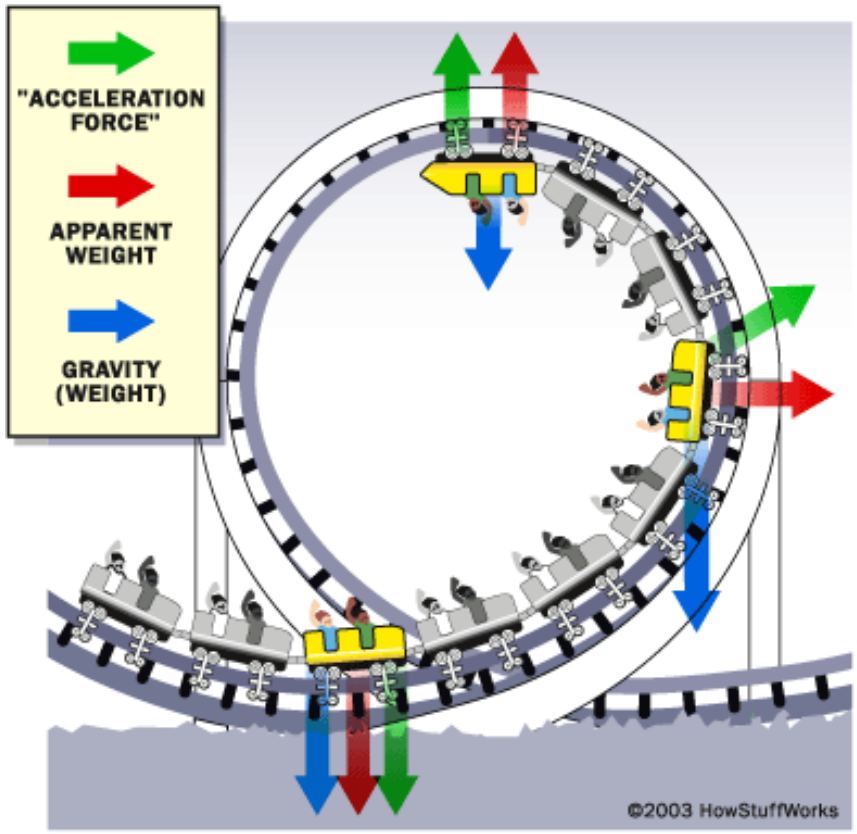
```
>>> x = x + 1
```

You can't use/update a variable that doesn't exist yet

```
>>> x = 0
>>> x = x + 1
```

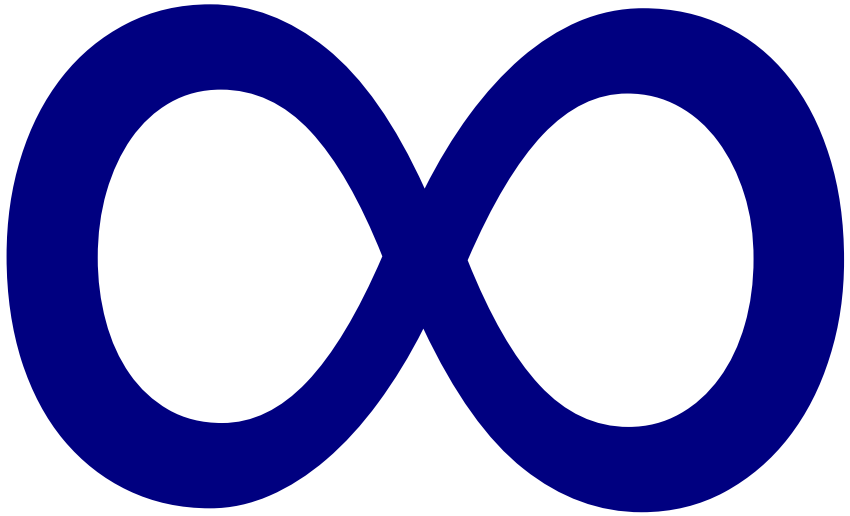
What happens in a loop?

1) Evaluate the condition, True or False
2) If the condition is false, exit the while statement and continue execution at the next statement
3) If the condition is true, execute the body and then go back to step 1



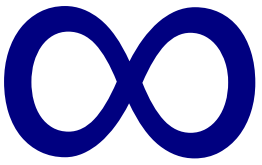
Infinite loops

- Try: bottles-of-beer-infinite-loop.py
- With CTRL+C you can stop the program



- Then try: bottles-of-beer-contains-error.py

Infinite loops



You need to make sure that the loop can end

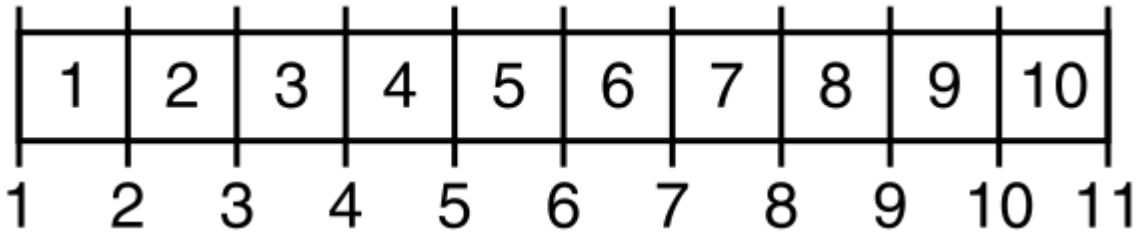
```
n = 99

while n>=0:
    if n > 0:
        print n, "bottles of beer on the wall,", n ,"bottles of beer."
        n = n - 1
        print "Take one down, pass it around,", n, "bottles of beer on the wall...\n"

    if n == 0:
        print "No more bottles of beer on the wall, no more bottles of beer."
        print "Go to the store and buy some more, 99 bottles of beer on the wall...\n"
```

Type of errors

- Syntax error (typos, forgetting closing quotes)
  - Python will give an error
- Logic error
  - seconds = hours / 3600
  - average = a + b / 2
- Off-by-one error



Off-by-one or fence-post error

Useful infinite loops

- Let a sensor measure the temperature endlessly
- Ask user for input until an answer is correct

```
while True:
    line = raw_input("Say 'when'> ")
    if line == 'when':
        break
    print line
print 'Done!'
```

say-when.py

continue and break

```
while True:
    line = raw_input('> ')
    if line[0] == '#' :
        continue
    if line == 'done':
        break
    print line
print 'Done!'
```

Example from the book
continue-and-break.py

for loops

applepie.py

```
groceries = ["apples", "flour", "butter", "eggs", "sugar"]

for item in groceries:

    print "Don't forget the", item
```

snoepjes.py

```
for i in range(10):

    print i, "Ik mag niet met snoepjes gooien"
```

Count stuff

```
total = 0

for i in [32, 45, 8, 10, 25]:
    total = total + i

print "Total:", total
```

Example from book
count-stuff.py

Maximum

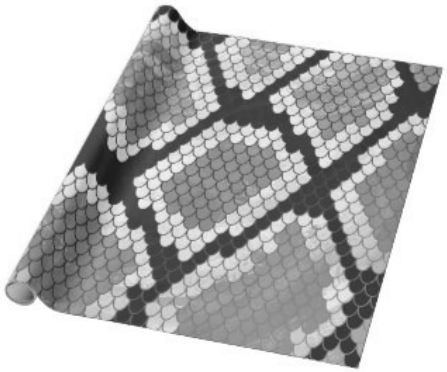
```
largest = None
print 'Before:', largest


for itervar in [3, 41, 12, 9, 74, 15]:
    if largest is None or itervar > largest :
        largest = itervar
    print 'Loop:', itervar, largest


print 'Largest:', largest
```

Example from book
get-maximum.py

Wrap-up



- A loop usually has the following
  - A variable is declared (counter or something finite)
  - Evaluation (conditional) of variable to end the loop
  - Something is done in the loop

Strings

Strings

- A string is a sequence

```
>>> fruit = “banana”
>>> letter = fruit[1]
```

b	a	n	a	n	a
0	1	2	3	4	5

Length and indices

- Length

```
>>> len(fruit)
```

- Get the last index

```
>>> length = len(fruit)
>>> last = fruit[length]
>>> last = fruit[length-1]
```

- Get the last letter

```
>>> fruit[-1]
```

Letter by letter: while loop

```
index = 0
while index < len(fruit):
    letter = fruit[index]
    print letter
    index = index + 1
```

Letter by letter: for loop

```
for letter in fruit:
    print letter
```

Slices

```
>>> s = “banana”
>>> s[0:3]   # ban
>>> s[:3]    # ban
>>> s[3:]    # ana
>>> s[3:3]   # ''
>>> s[2:5]   # nan
>>> s[:]     # banana
```

Mutability

```
>>> x = "cookies"
```

This will give an error:

```
>>> x[0] = “b”
```

Solution, use slices:

```
>>> new = x[:3] + "l"
>>> print new
```

Count letters

- Write a program that counts all the a's in 'banana'

- Hint
  - for loop
  - a variable to count
  - if statement to check if the letter is an a

“in” and string comparison

- Check if letter or substring is in a string

```
>>> 'a' in 'banana'
>>> 'ana' in 'banana'
```

- Comparisons

```
>>> 'mouse' == 'mouse'
>>> 'Mouse' == 'mouse'
>>> 'elephant' > 'cow'
>>> 'dinosaur' < 'cat'
>>> 'mouse' > 'Zebra'
```

String methods

- upper() / lower()
- find()
- startswith()
- strip()

How to use this:

```
>>> s = 'cow'
>>> s.upper()
>>> type(s)
>>> dir(s)
>>> help(str.upper)
```

Parsing strings

```
>>> s = ' From: Barbera <barbera@van-schaik.org>'
>>> s.find("@")
23
>>> start = s.find("<")
>>> end = s.find(">")
>>> s[start:end]
'<barbera@van-schaik.org'
>>> s[start+1:end]
'barbera@van-schaik.org'
```

More

- Chapter 6
  - Format operator
- Documentation string methods
  - https://docs.python.org/2/library/string.html

Assignment – guessing game

- Write a program that generates a random number between 1 and 10
- Let the user guess which number the computer had in mind
- The user can guess at most 5 times
- Give hints “higher” and “lower” when the guessed number is too high or too low
- Partial code is available on the wiki

guess-game.py

Next week

- Next: Files and lists
- More programming exercise?
  - Chapter 5 and 6 of the book
- Tot volgende week!!