while loop

while loop is another way to repeat some fragment of code. The loop is repeated as long as some condition is True.

In [ ]:
```
i = 0
while i < 5:
    print(i)
    i = i + 1
```

break / continue

The execution of both loop (for and while) can be interrupted using **break** and **continue**. Analyze examples below to figure out how they behave.

In [ ]:
```
print("continue")
i = 0
while i < 5:
    i += 1
    if i == 2:
        continue
    print(i)

print("break")
i = 0
while i < 5:
    i += 1
    if i == 2:
        break
    print(i)
```

while else

You can add else after while loop. The code will execute once when the condition is no longer True. It will not execute when you break the loop.

In [ ]:
```
while True:
    break
else:
    print("This will not be printed!")

i = 0
while i < 2:
    print(i)
    i += 1
else:
    print("Hello one time!")
```

for else

You can also add else after for loop. The code will execute if you do NOT use break.

In [ ]:
```
for i in [1,2,3]:
    break
else:
    print("This will not be printed!")

for i in [1,2,3]:
    print(i)
else:
    print("Hello one time!")
```

Processing all elements and creating list - map and list comprehension

In [ ]:
```
l = ['1', '2', '3']

print("List comprehension")
l1 = [int(x) for x in l]
print(l1)
l2 = [a*a for a in l1]
print(l2)

print("Map")
lm1 = list(map(int, l))
print(lm1)
lm2 = list(map(lambda x: x*x, l1))
print(lm2)
```

Exercises

Ex. 1. Rewrite the following code to use while instead of for.

In [ ]:
```
for i in range(1, 20, 2):
    print(i)
```

In [ ]:

Ex. 2. Write a code that will echo user input until he writes 'quit'.

In [ ]:

Ex. 3. Write a program that will read integer N and prints numbers from 1 to m, where sum(1, ..., m) < N.

In [ ]:

Ex. 4. Create a list with some number of random integers from 1 to 100. You should stop adding new integers to the list when the maximum element in the list is smaller than sum of all the numbers without the maximum.

In [ ]:
```
import random
print(random.randint(1, 100))
```

Ex. 5. Modify the my_sqrt(n, i) funtion from Lab 3 to end when specific approximation is reached. It means, you should not repeat the algorithm specific number of times, but end it when the elements you have to add are smaller than some **epsilon**.

In [ ]:
```
def my_sqrt(n, eps):
```

Ex. 6. Use while loop to print the binary representation of a number.

In [ ]:

Ex. 7. Process each element using map function on provided list to obtains list of lists containing number and its square.

In [ ]:

Ex. 8. Process each element using list comprehension on provided list to obtains list of lists containing number and its square.

In [ ]:

Ex. 9. Use any form of iteration and split() function to read unspecified number of space-separated integers from input().

In [ ]:

Extra - Image processing

Below is an example of code that process the image. To use it you may have to install cImage:

```
pip install cImage
```

If code below successfully import image, but fails on loading image with image.FileImage you may have wrong image module installed. Try reinstalling with the following commands:

```
pip uninstall cImage
pip uninstall PIL
pip uninstall Pillow
pip uninstall image
pip install cImage
```

1. In the example above we remove red colour from image. Analyze the code and try to write some other filters
   - convert to greyscale
   - convert to black-white
   - convert to sepia
2. Try to make the image 2 times bigger.
3. Try to detect edges using Sobel detection algorithm (image.png may be better for this)

In [ ]:
```
import image


img =image.FileImage("LutherBellPic.jpg")                   # we load the image
print(img.getWidth(), img.getHeight())                      # we can check sizes with those functions
newimg = image.EmptyImage(img.getWidth(), img.getHeight())  # we create new image with appropriate size
s
print(img.getPixel(0, 0))                                   # we can get each pixel using its position
print(img.getPixel(0, 0).getBlue())                         # pixel is represented as RGB. we can get e
ach color

for col in range(img.getWidth()):
    for row in range(img.getHeight()):
        p = img.getPixel(col, row)

        newred = 0
        green = p.getGreen()
        blue = p.getBlue()

        newpixel = image.Pixel(newred, green, blue)         # we create pixel with some values
        newimg.setPixel(col, row, newpixel)                 # and set this pixel in appropriate locatio
n in new image

newimg.save('bell_without_red.jpg')                         # we save new file
```

In [ ]: