Advanced Numbers

In this lecture we will learn about a few more representations of numbers in Python.

Hexadecimal

Using the function `hex()` you can convert numbers into a hexadecimal format:

```
In [1]:   hex(246)

Out[1]:   '0xf6'
```

```
In [2]:   hex(512)

Out[2]:   '0x200'
```

Binary

Using the function `bin()` you can convert numbers into their binary format.

```
In [3]:   bin(1234)

Out[3]:   '0b10011010010'
```

```
In [4]:   bin(128)

Out[4]:   '0b10000000'
```

```
In [5]:   bin(512)

Out[5]:   '0b1000000000'
```

Exponentials

The function `pow()` takes two arguments, equivalent to x^y. With three arguments it is equivalent to `(x^y)%z`, but may be more efficient for long integers.

```
In [6]:   pow(3,4)

Out[6]:   81
```

```
In [7]:   pow(3,4,5)

Out[7]:   1
```

Absolute Value

The function `abs()` returns the absolute value of a number. The argument may be an integer or a floating point number. If the argument is a complex number, its magnitude is returned.

```
In [8]:  abs(-3.14)

Out[8]:  3.14
```

```
In [9]:  abs(3)

Out[9]:  3
```

Round

The function `round()` will round a number to a given precision in decimal digits (default 0 digits). It does not convert integers to floats.

```
In [10]:  round(3,2)

Out[10]:  3
```

```
In [11]:  round(395,-2)

Out[11]:  400
```

```
In [12]:  round(3.1415926535,2)

Out[12]:  3.14
```

Python has a built-in math library that is also useful to play around with in case you are ever in need of some mathematical operations. Explore the documentation here!