

# Variable type dictionary I

- To store data in format key:value => use key to access value

```
1 my_dict = { 'chr1' : 'gene1', 'chr2' : 'gene2' }  
2 print(my_dict)  
3 print(my_dict['chr1'])
```

```
{'chr1': 'gene1', 'chr2': 'gene2'}  
gene1
```

- Keys must be unique => no duplicated entries!

```
1 my_dict = { 'chr1' : 'gene1', 'chr1' : 'gene51510' }  
2 print(my_dict)
```

```
{'chr1': 'gene51510'}
```

- Instead put a list inside a dictionary

```
1 my_dict = { 'chr1' : ['gene1', 'gene51510'] }  
2 print(my_dict['chr1'][0])  
3 print(my_dict['chr1'][1])
```

```
☞ gene1  
gene51510
```

# Variable type dictionary II

- All data types are allowed!

```
1 my_dict = { 'chr1' : ['gene1', 'gene51510'],  
2             'chr2' : 120500,  
3             'chr3' : False,  
4             0 : 1,  
5             False : True }  
6 print(my_dict)  
7 print(my_dict['chr1'][0])  
8 print(my_dict['chr1'][1])  
9 print(my_dict['chr2'])  
10 print(my_dict[0])  
11 print(my_dict[False])
```

```
{'chr1': ['gene1', 'gene51510'], 'chr2': 120500, 'chr3': False, 0: True}  
gene1  
gene51510  
120500  
True  
True
```

# Variable type dictionary III

- Beginning with Python 3.7 dictionaries are ordered; dictionaries in earlier Python versions are unordered
- How many items are in the dictionary (or in a list)?



```
1 my_dict = { 'chr1' : ['gene1', 'gene51510'] }  
2 print(len(my_dict))  
3 print(len(my_dict['chr1']))
```

1  
2

# Gap Exercises 1

- Access GitLab exercise folder
- Fill in/correct the code of the dictionary exercises of 'gap\_exercise1\_basics.ipynb'

# DNA, RNA, and peptide sequences

# Reverse complement

What happens here?

```
1 def revcomp( seq ):
2
3     seq = seq.lower()
4
5     #key:value (=dictionary)
6     complement = { 'a':'t', 't':'a', 'c':'g', 'g':'c' }
7
8     new_seq = []
9
10    for nt in seq:
11        new_seq.append( complement[ nt ] )
12
13    #list[::-1] inverts list (last element becomes first)
14    new_seq = "".join( new_seq[::-1] )
15
16    return new_seq
```

Sequence of bases e.g. ATGACATGA

Converts input to lower case: atgacatga

Get complement for each base

Inverts list (=reverse)

# Exercises F – Part1

- 1.1) Write a function to get the reverse complement (upper case letters) of a DNA sequence given in upper case letters!
- 1.2) Write a function to convert a DNA sequence into a RNA sequence!
- 1.3) Write a function to translate a DNA sequence into amino acids (first frame only)!

Tipp: [http://en.wikipedia.org/wiki/DNA\\_codon\\_table](http://en.wikipedia.org/wiki/DNA_codon_table)

- 1.4) Write a function to translate DNA sequences in all 6 frames into peptide sequences! The longest peptide sequence per DNA sequence should be returned!