



Additional Slides week 2 - Lasse Fierz

Computer Science II

Wednesday, March 1, 2023

No guarantee of correctness

1. String Formatting

Old String Formatting

Old way of formatting strings:

"Normal, comma separated"

```
name = "Peter"  
print("Hello", name) #Hello Peter
```

Personally I still use this for quick prints.

String addition

```
print("Hello " + name) #Hello Peter
```

% Operator Formatting (Old way)

```
print("Hello \% s" \% "Peter") #Hello Peter
```

Achtung wichtig: Without the \ (Latex stuff...)
Here for strings %s but for e.g. int %d

Outdated because it's annoying for long strings with many inputs.

String.format (python 3+)

string.format

```
name = "Peter"  
age = 20  
print("Hello {} your age is {}".format(name,age))  
# Hello Peter your age is 20
```

Also possible with indexes:

```
name = "Peter"  
age = 20  
print("Hello {1} your age is {0}".format(age,name))  
# Hello Peter your age is 20
```

fstring Formatting (Python 3.6+)

fstring

```
name = "Peter"  
age = 20  
printf("Hello {name} your age is {age}")  
# Hello Peter your age is 20
```

fstrings allow for very powerful formatting:

```
a = 1  
b = 3  
printf("a + b is {a+b}.") # a + b is 4.
```

For more advanced usecases check the official documentation.
Note that any method mentioned here is still valid however best practice would be to use fstrings or `.format`.