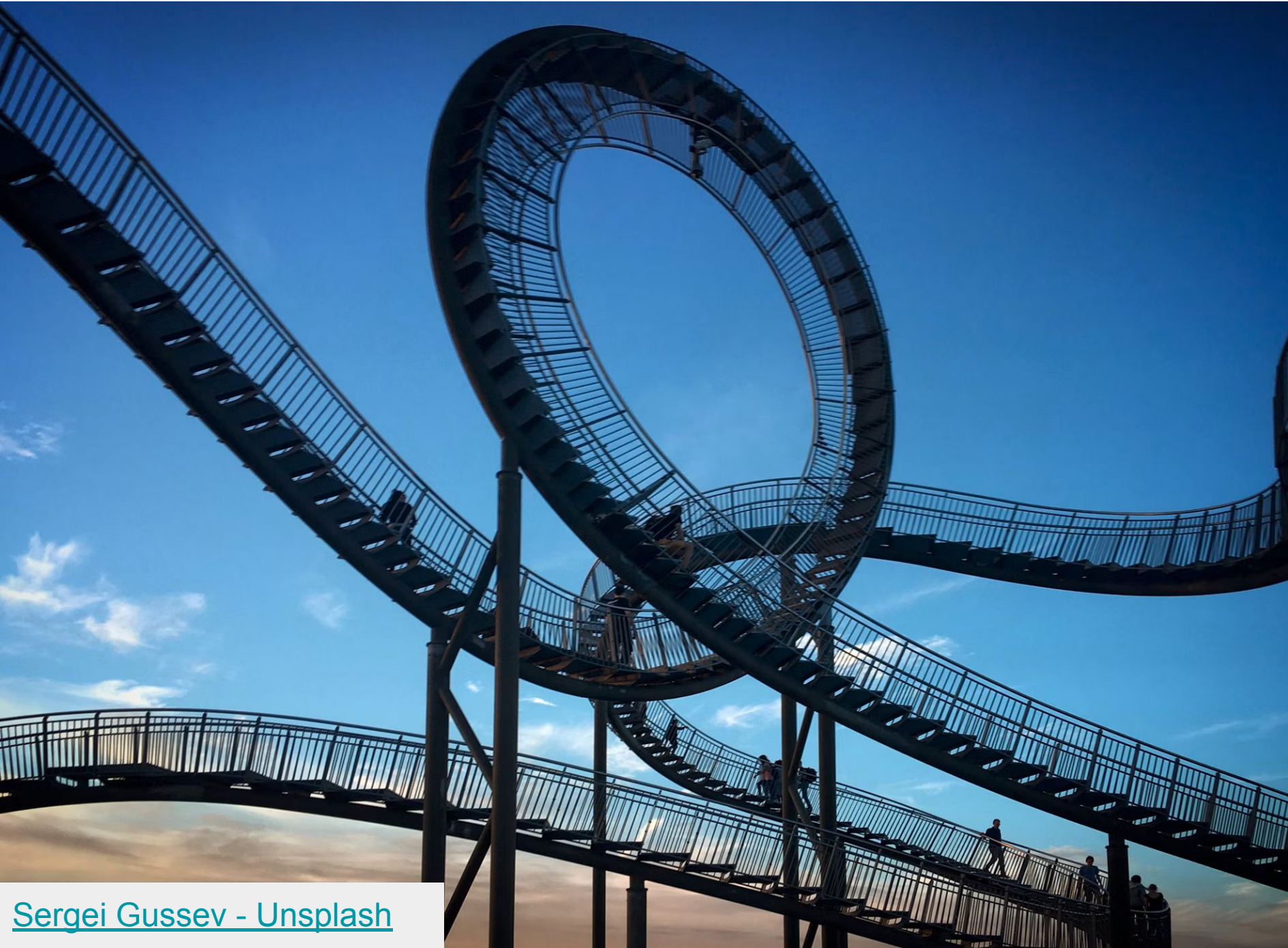


# bash scripting – loops

Adapted from materials by Dr. Carrier



# for loops

Do we expect C-style or Python-style for loops?

# for loops

Do we expect C-style or Python-style for loops?

```
for(int i = 0; i < 10; i++){ ... }
```

vs

```
for i in range(10):
```

for loops

More like Python!

# for loops

More like Python!

```
for var in list  
do  
    ...  
done
```

# for loops

More like Python!

```
for var in list
do
    ...
done
```

```
for var in 1 2 3 4 5
for var in "alice" "bob" "carol"
for idx in {0..10..2}
for var
for var in $@
```

# A note on syntax

`$(command)` will run the command inside

# A note on syntax

`$(command)` will run the command inside

```
var=$(ls)
```

```
for idx in $(seq 0 10)
```

```
do
```

```
...
```

```
done
```



# while loops

```
while condition  
do  
    ...  
done
```

# What are our conditions? Same as if

Don't worry about memorizing these:

<https://ryanstutorials.net/bash-scripting-tutorial/bash-if-statements.php>

Operator	Description
<b>! EXPRESSION</b>	The EXPRESSION is false.
<b>-n STRING</b>	The length of STRING is greater than zero.
<b>-z STRING</b>	The length of STRING is zero (ie it is empty).
<b>STRING1 = STRING2</b>	STRING1 is equal to STRING2
<b>STRING1 != STRING2</b>	STRING1 is not equal to STRING2
<b>INTEGER1 -eq INTEGER2</b>	INTEGER1 is numerically equal to INTEGER2
<b>INTEGER1 -gt INTEGER2</b>	INTEGER1 is numerically greater than INTEGER2
<b>INTEGER1 -lt INTEGER2</b>	INTEGER1 is numerically less than INTEGER2
<b>-d FILE</b>	FILE exists and is a directory.
<b>-e FILE</b>	FILE exists.
<b>-r FILE</b>	FILE exists and the read permission is granted.
<b>-s FILE</b>	FILE exists and its size is greater than zero (ie. it is not empty).
<b>-w FILE</b>	FILE exists and the write permission is granted.
<b>-x FILE</b>	FILE exists and the execute permission is granted.