

# Lab 1 - Basic R

Operators, Variables, Expressions, If/Else statements, While/For loops

March 26th, 2018

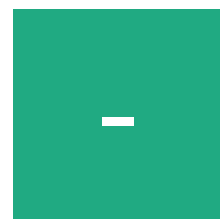
# basic operators



addition

> 5+5

[1] 10



subtraction

> 5-5

[1] 0



multiplication

> 5\*5

[1] 25



division

> 5/5

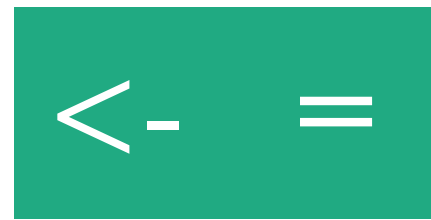
[1] 1



exponentiation

> 5^5

[1] 3125



# Variables

integer (numeric)

```
> number <- 6  
> number  
[1] 6
```

double (numeric)

```
> double <- 6.6  
> double  
[1] 6.6
```

string/character

```
> name <- "George"  
> name  
[1] "George"
```



# Vectors

creating vectors

```
> shopping <- c("bananas", "oranges", "peanuts")  
> shopping  
[1] "bananas" "oranges" "peanuts"
```

indexing vectors

```
> shopping[2]           > shopping[1:2]           > shopping[-2]  
[1] "oranges"           [1] "bananas" "oranges"   [1] "bananas" "peanuts"
```

manipulating vectors

```
> shopping[1] <- "peaches"  
> shopping  
[1] "peaches" "oranges" "peanuts"
```

summing vectors

```
> numbers <- c(1, 2, 3)  
> total <- sum(numbers)  
> total  
[1] 6
```

# logical expressions

==

are the two sides of **one** expression equal?

> <

is one side of **one** expression 'larger'?

&&

**and**: are both **expressions** TRUE?

||

**or**: is at least one **expression** TRUE?

# control flow - if/else

if

TRUE

{ do something }

```
if(10 > 5){  
    print("10 is greater than 5")  
} else {  
    print("10 is not greater than 5")  
}
```

else

(if NOT true)

{ do something else }

[1] "10 is greater than 5"

# control flow - for loops

for

some **item** in a **list**

{ do something }

```
for(i in 1:10){  
  print(i)  
}
```

**Important** - the fact that we used **i** **\*does not not matter\***. You can use any variable name in place of **i**. Think of **i** as a 'placeholder'.

```
[1] 1  
[1] 2  
[1] 3  
[1] 4  
[1] 5  
[1] 6  
[1] 7  
[1] 8  
[1] 9  
[1] 10
```

# control flow - while loops

while

**something** is **TRUE**

{ do something }

{ make sure that **something** is eventually **FALSE**}

```
i <- 1
while(i <= 5){
  print(i)
  i <- i + 1
}
```

[1] 1

i is 1. less than 5, print it. Add 1 to i.

[1] 2

i is 2. less than 5, print it. Add 1 to i.

[1] 3

i is 3. less than 5, print it. Add 1 to i.

[1] 4

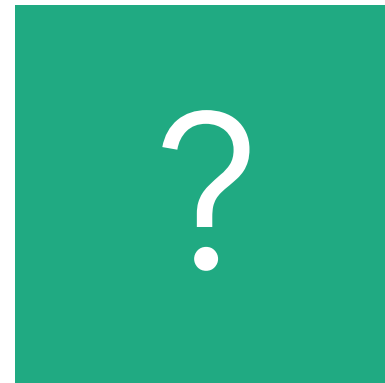
i is 4. less than 5, print it. Add 1 to i.

[1] 5

i is 5. equal to 5, print it. Add 1 to i.

i is 6. **not less than** 5. do nothing.





**questions?**