



# FS CONCEPTS AT A GLANCE

## Introduction



### GIT-ing your challenges done...

`git clone [url]`  
to copy the assignment from the repository.

`ls`  
allows you to see the content of your directory

`pwd`  
tells you the directory that you are in

`cd ..`  
allows you to move up to parent directory

`cd [file]`  
takes you to the directory you are working on.

`git add [file]`  
adds the file as it looks now to your next commit

`git commit -m "[descriptive message]"`  
commit your staged content as a new commit snapshot

`git push`  
Transmit local branch commits to the remote repository branch  
(AKA submits your challenge so we can see your work)

### DATA TYPES

#### PRIMITIVE DATA TYPES

string

```
let firstName = "Hedy";  
let lastName = "Lamar";
```

number

```
let age = 10;  
let amount = 20.50;
```

Boolean

```
let eaten = true;  
let done = false;
```

#### SPECIAL DATA TYPES

```
/*  
This value is declared with a name  
but the value is undefined  
*/  
let box;
```



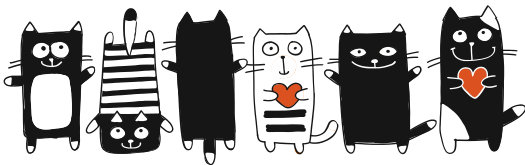
Empty (undefined) variable declared by the name "box"

```
let box2 = null;
```



Null is like a vacuum. It's there but it's nothing.

### ARRAYS



```
const catArray = ["Tabby", "Stripes",  
"Shadow", "Cutie", "Fishy", "Kitty"]
```

```
catArray.length = 6;
```

```
catArray[0] = "Tabby";  
catArray[1] = "Stripes";  
catArray[2] = "Shadow";  
catArray[3] = "Cutie";  
catArray[4] = "Fishy";  
catArray[5] = "Kitty";
```

The index tells you the position of the element in an array.

### OBJECTS



```
const pet = {  
  name: "Buddy",  
  breed: "bulldog",  
  food: "Alpo",  
}
```

```
pet.name = "Buddy";  
pet.breed = "bulldog"
```

### COMPARISON OPERATORS

=== equal to  
!== not equal to  
> greater than  
< less than  
>= greater than or equal to  
<= less than or equal to

### LOGICAL OPERATORS

&& and  
|| or  
! not

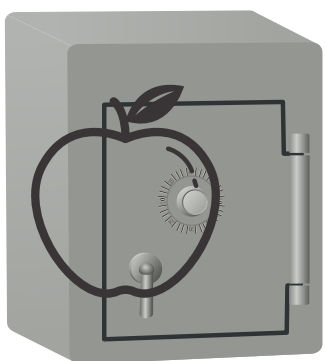
### var , const & let



`var fruit =`

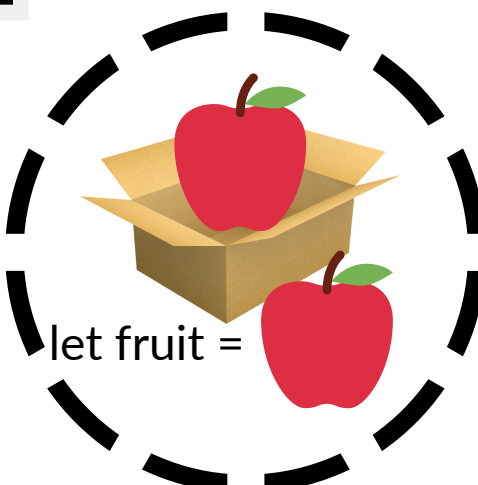
`fruit =`

You can swap the apple with an orange later on.



`const fruit =`

The apple is locked in a safe and you can never, ever swap it.



`let fruit =`

You can swap the apple later on within the force field..