EXAMPLE 24: UNDERSTANDING `typeof`

EXAMPLE 24: UNDERSTANDING `typeof`

JAVASCRIPT HAS A HELPFUL OPERATOR CALLED `typeof` THAT HELPS YOU CHECK THE TYPE OF AN OBJECT.

THIS IS QUITE USEFUL, BECAUSE JAVASCRIPT IS NOT STRONGLY TYPED, I.E. YOU DON'T HAVE TYPE DECLARATIONS TO GO BY

EXAMPLE24: UNDERSTANDING typeof

```
console.log(typeof "123");
console.log(typeof 123);
console.log(typeof undefined);
console.log(typeof []);
console.log(typeof true);
```

EXAMPLE 24: UNDERSTANDING typeof

```
console.log(typeof "123");
console.log(typeof
console.log(typeof undefined);
console.log(typeof []);
console.log(typeof true);
```

```
string
```

EXAMPLE 24: UNDERSTANDING typeof

```
console.log(typeof "123");
console.log(typeof 123);
```

```
number
```

```
console.log(typeof undefined);
console.log(typeof []);
console.log(typeof true);
```

EXAMPLE 24: UNDERSTANDING typeof

```
console.log(typeof "123");
console.log(typeof 123);
console.log(typeof undefined);
console.log(typeof []);
console.log(typeof true);
```

```
undefined
```

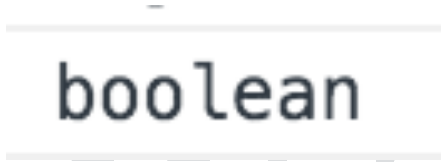
EXAMPLE 24: UNDERSTANDING typeof

```
  console.log(typeof
"123");
  console.log(typeof 123);
  console.log(typeof
undefined);
                                        object
  console.log(typeof []);
  console.log(typeof
true);
```

WOW! ARRAYS ARE OBJECTS TOO!

EXAMPLE 24: UNDERSTANDING typeof

```
console.log(typeof "123");
console.log(typeof 123);
console.log(typeof undefined);
console.log(typeof
console.log(typeof true);
```



```
boolean
```

EXAMPLE 24: UNDERSTANDING `typeof`

ONE LITTLE NOTE - ANY TYPE OF OBJECT WILL SIMPLY RETURN `object`

`typeof` IS NOT SMART ENOUGH TO DISTINGUISH BETWEEN DIFFERENT TYPES OF OBJECTS.

`instanceof` IS THE SOLUTION TO THIS PROBLEM:-)