Optional chaining

Optional chaining

There is this problem that if the object that you are accessing might or might not contain an object attribute say `address` and you can trying to access `address`'s attribute further say `user.address.street`, then your JavaScript will crash because it is trying to access an `undefined`'s attribute.

```
// The problem
let user = {};


user.address.street // Error
```

To solve this we have an operator called `optional chaining`, the `?.` operator.

So essentially, you replace the dot notation with `?.` and it will stop the evaluation if the value before `?.` is `undefined/null` and finally returns `undefined`.

```
value?.prop


// Will be equal to value.prop if value exists
// Otherwise if value is undefined/null it will return undefined.
```

This is a much safer way of accessing attributes that you know will be undefined.

Don't overuse optional chaining

You should only use the optional chaining operator only when it's ok that something doesn't exist

In addition, the variable that you call optional chaining on must be declared otherwise, it will be an error.

```
user?.address // If there is not let user; then it will be an error
```

Revision #2
Created 2022-12-20 17:46:26 UTC by Tamarine
Updated 2022-12-20 18:06:50 UTC by Tamarine