

Errors and Debugging

1. Errors usually occur when:
 - a. You don't write valid JS code.
 - b. Your code is valid JS but the code will be trying to do something not allowed.
2. There are mainly 4 types of errors:

a. Syntax Error

- i. This occurs when there is something wrong with your syntax.
- ii. Syntax errors prevent your code from running at all.
- iii. Example:

```
function myFunction() {  
  console.log("hello world")  
}
```

- iv. In the code above, the keyword *"function"* is spelled wrongly.
- v. So that is an example of a syntax error.

b. Runtime Error

- i. This type of error occurs while the code is running.
- ii. Example:

```
let val=true;  
function myFunction() {  
  while(val) {  
    console.log("hello world");  
  }  
}
```

- iii. This code above is an infinite loop.
- iv. This might cause the browser to freeze because the code will keep running forever.
- v. So this is an example of a runtime error.

c. Type Error

- i. Occurs when you try to use a type the wrong way.
- ii. Example:

```
let name="code";  
  
console.log(name());
```

- iii. In the above code, we are trying to invoke the variable **"name"**
- iv. But the type of **"name"** *string* and you can't invoke a *string*.
- v. So this is an example of a type error.

d. Reference Error

- i. This type of error occurs when you try to access a variable that does not exist in the current scope.
- ii. Example:

```
function myFunction() {  
    let name="code";  
}  
  
console.log(name);
```

- iii. In the above code, the scope of the variable **"name"** is inside the function only.
- iv. And we are trying to access the variable outside the function.
- v. Since the variable does not exist in the global scope, trying to access it in the global scope will result in a Reference Error.