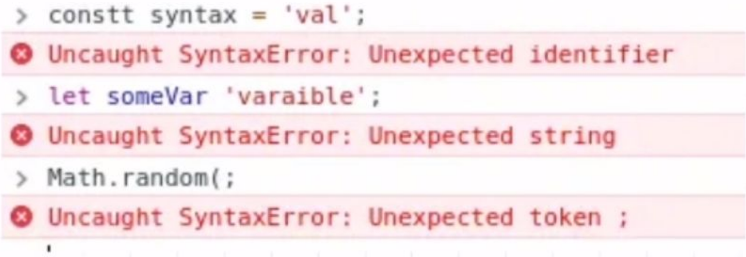
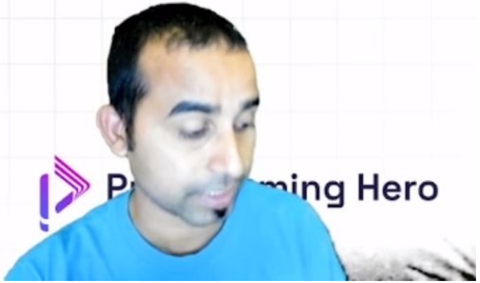
Syntax Error

Syntax error is triggered when you write code that is not syntactically correct.

- Missing inverted commas
- Missing closing parentheses
- Improper alignment of curly braces or other characters



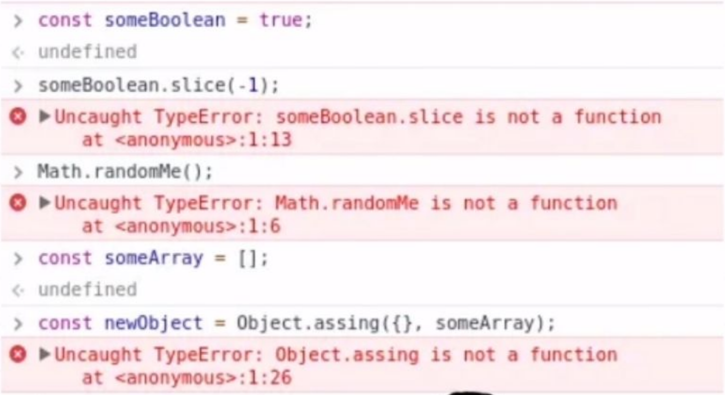
```
> constt syntax = 'val';
⊗ Uncaught SyntaxError: Unexpected identifier
> let someVar 'varaible';
⊗ Uncaught SyntaxError: Unexpected string
> Math.random(;
⊗ Uncaught SyntaxError: Unexpected token ;
```



Type Error

Type error is created when some value doesn't turn out to be of a particular expected type.

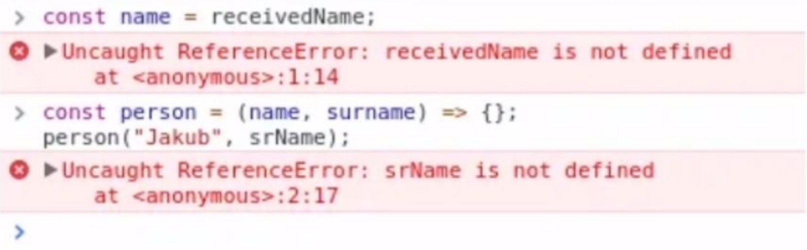
- Calling objects that are not methods.
- Attempting to access properties of null or undefined objects
- Treating a string as a number or vice versa



```
> const someBoolean = true;
< undefined
> someBoolean.slice(-1);
✖ ▶Uncaught TypeError: someBoolean.slice is not a function
      at <anonymous>:1:13
> Math.randomMe();
✖ ▶Uncaught TypeError: Math.randomMe is not a function
      at <anonymous>:1:6
> const someArray = [];
< undefined
> const newObject = Object.assing({}, someArray);
✖ ▶Uncaught TypeError: Object.assing is not a function
      at <anonymous>:1:26
```

Reference Error

- forgotten to define a value for the variable before using it
- we might be trying to use an inaccessible variable in our code.
- Making a typo in a variable name.
- Trying to access block-scoped variables outside of their scopes.



```
> const name = receivedName;
✖ ▶Uncaught ReferenceError: receivedName is not defined
      at <anonymous>:1:14
> const person = (name, surname) => {};
  person("Jakub", srName);
✖ ▶Uncaught ReferenceError: srName is not defined
      at <anonymous>:2:17
>
```

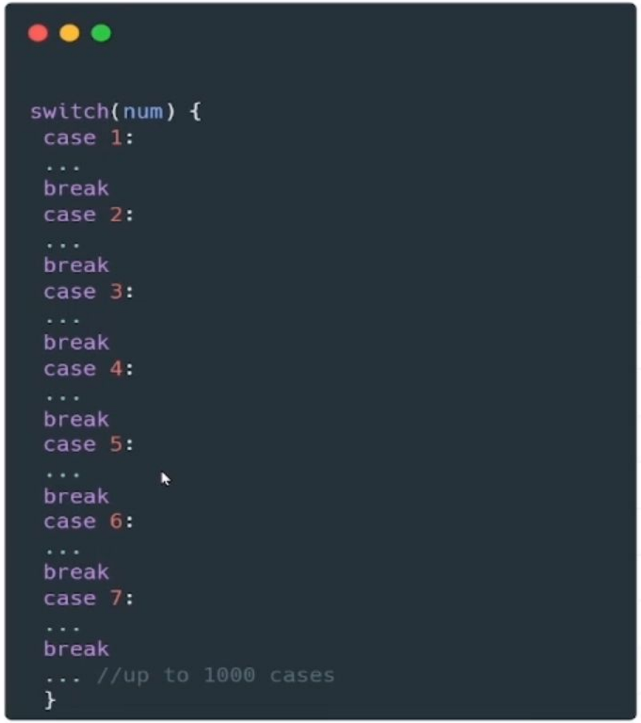
Internal Error

The Internal Error occurs when an exception occurs in the JavaScript runtime engine. It may or may not indicate an issue with your code.

InternalError occurs in two scenarios only:

- When a patch or an update to the JavaScript runtime carries a bug that throws exceptions
- When your code contains entities that are too large for the JavaScript engine (e.g. **too many switch cases, too large array initializers, too much recursion**)

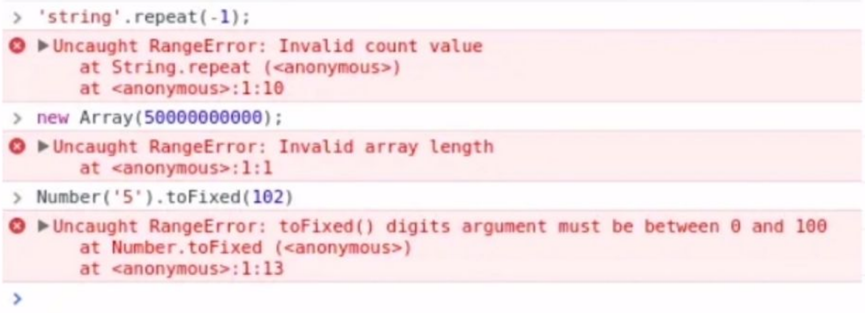
Internal Error Example



```
switch(num) {
 case 1:
 ...
 break
 case 2:
 ...
 break
 case 3:
 ...
 break
 case 4:
 ...
 break
 case 5:
 ...
 break
 case 6:
 ...
 break
 case 7:
 ...
 break
 ... //up to 1000 cases
 }
```

Range Error

RangeError —thrown when a value is not in an allowed range



```
> 'string'.repeat(-1);
⊗ ▶Uncaught RangeError: Invalid count value
      at String.repeat (<anonymous>)
      at <anonymous>:1:10
> new Array(50000000000);
⊗ ▶Uncaught RangeError: Invalid array length
      at <anonymous>:1:1
> Number('5').toFixed(102)
⊗ ▶Uncaught RangeError: toFixed() digits argument must be between 0 and 100
      at Number.toFixed (<anonymous>)
      at <anonymous>:1:13
>
```

Interview Questions

1. What is syntax error?
2. What type of error will you see? How will you fix it?

```
function errors(a){
  if(a > 5){
   return true
  } else {
   return false



 console.log(errors(5))
```

Interview Questions

3. When do we get reference error?

4. What type of an error will this code generate? How will you fix it?

```
function text() {
    var str = "Hello World";
    return str;
}
```

5. a) What's wrong with the code below?

```
var foo;
console.log(foo.bar);
```

b) What type of error will the code generate?

c) What is the correct way to write the code?

SLIDE

Thank you!

PROGRAMMING HERO