

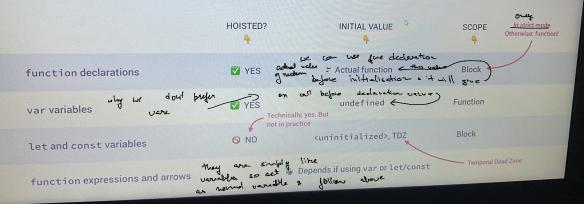
OISTING IN JAVASCRIPT

Hoisting: Makes some types of variables accessible/usable in the code before they are actually declared. "Variables lifted to the top of their scope".

BEHIND THE SCENES

Before execution, code is scanned for variable declarations, and for each variable, a new property is created in the **variable environment object**.





TEMPORAL DEAD ZONE, LET AND CONST

```
const myName = 'Jonas';

if (myName === 'Jonas') {
  console.log('Jonas is a ${job}');
  const age = 2037 - 1989;
  console.log(age);
  const job = 'teacher';
  console.log(x);
}
```

TOZ

TEMPORAL DEAD ZONE FOR job VARIABLE

Different kinds of error messages:

ReferenceError: Cannot access 'job' before initialization
ReferenceError: x is not defined

WHY HOISTING?

- Using functions before actual declaration;
- var hoisting is just a byproduct

WHY TDZ?

- Makes it easier to avoid and catch errors: accessing variables before declaration is bad practice and should be avoided;
- Makes const variables actually work