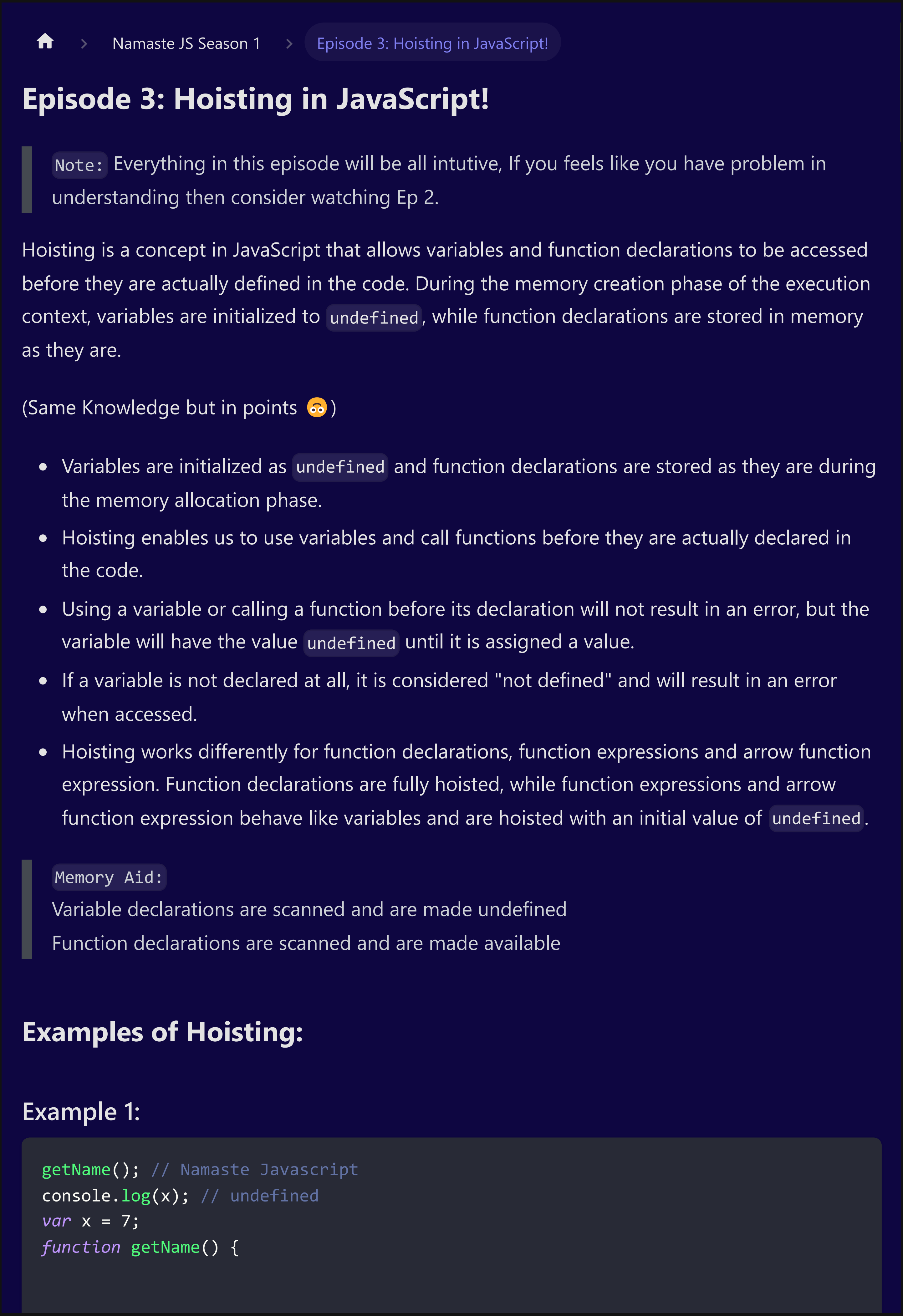


Episode 3: Hoisting in JavaScript!

> `Note:` Everything in this episode will be all intutive, If you feels like you have problem in understanding then consider watching Ep 2.

Hoisting is a concept in JavaScript that allows variables and function declarations to be accessed before they are actually defined in the code. During the memory creation phase of the execution context, variables are initialized to `undefined`, while function declarations are stored in memory as they are.

(Same Knowledge but in points 🙃)

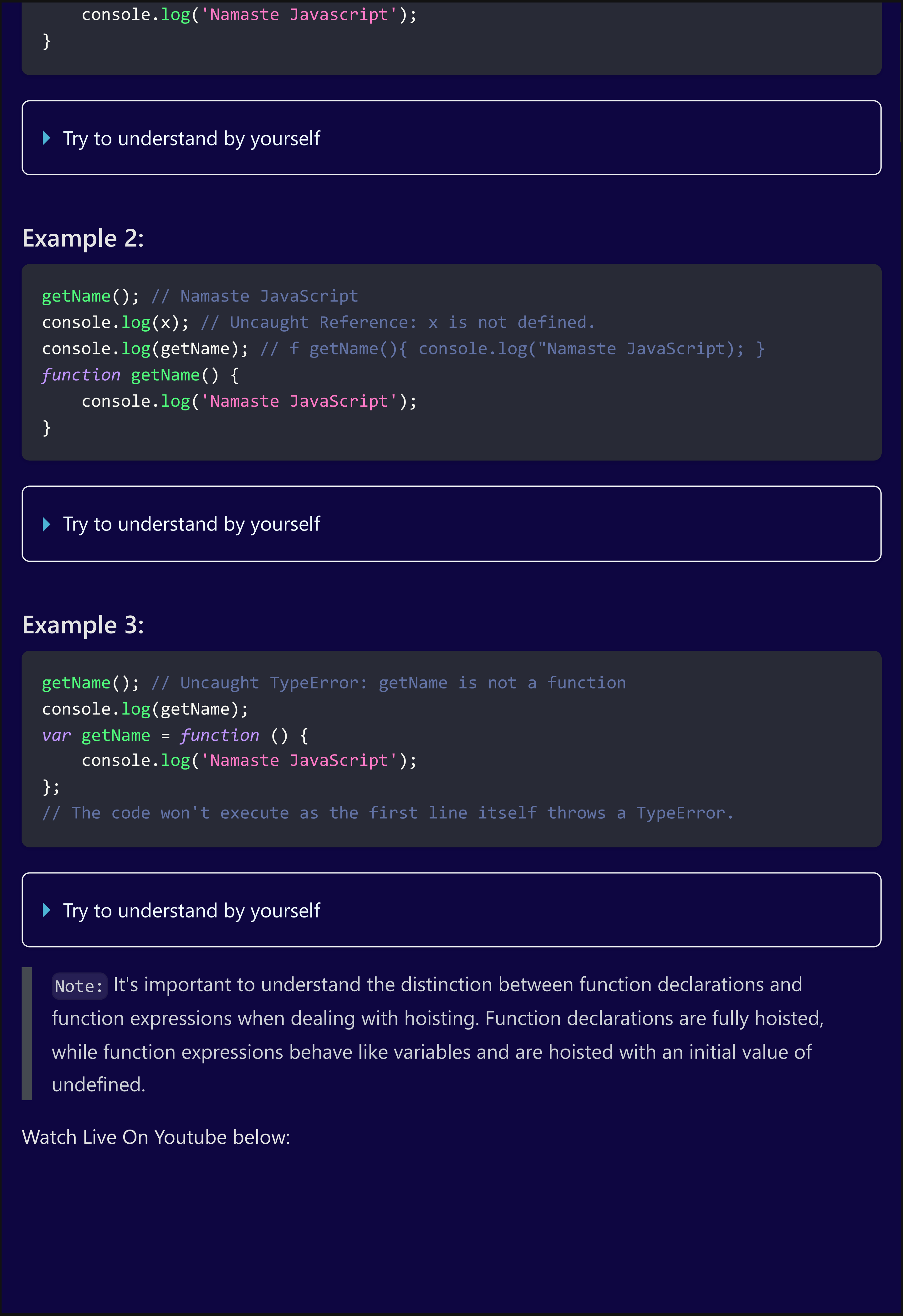
- Variables are initialized as `undefined` and function declarations are stored as they are during the memory allocation phase.
- Hoisting enables us to use variables and call functions before they are actually declared in the code.
- Using a variable or calling a function before its declaration will not result in an error, but the variable will have the value `undefined` until it is assigned a value.
- If a variable is not declared at all, it is considered "not defined" and will result in an error when accessed.
- Hoisting works differently for function declarations, function expressions and arrow function expression. Function declarations are fully hoisted, while function expressions and arrow function expression behave like variables and are hoisted with an initial value of `undefined`.

> `Memory Aid:`
> Variable declarations are scanned and are made undefined
> Function declarations are scanned and are made available

Examples of Hoisting:

Example 1:

```
getName(); // Namaste Javascript
console.log(x); // undefined
var x = 7;
function getName() {
```



```
    console.log('Namaste Javascript');
}
```

▸ Try to understand by yourself

Example 2:

```
getName(); // Namaste JavaScript
console.log(x); // Uncaught Reference: x is not defined.
console.log(getName); // f getName(){ console.log("Namaste JavaScript); }
function getName() {
    console.log('Namaste JavaScript');
}
```

▸ Try to understand by yourself

Example 3:

```
getName(); // Uncaught TypeError: getName is not a function
console.log(getName);
var getName = function () {
    console.log('Namaste JavaScript');
};
// The code won't execute as the first line itself throws a TypeError.
```

▸ Try to understand by yourself

> `Note:` It's important to understand the distinction between function declarations and function expressions when dealing with hoisting. Function declarations are fully hoisted, while function expressions behave like variables and are hoisted with an initial value of undefined.

Watch Live On Youtube below:

✎ Edit this page