Title: What is a constant?

Answer:

Many programming languages feature *constants*, which are values that cannot be modified or changed. In JavaScript, the `const` identifier is modelled after constants, and the values assigned to a `const` cannot be reassigned.

In the example below, we initialize the variable `SPECIES` as a constant with the `const` keyword. Trying to reassign the variable will result in an error.

```
// Assign value to const
const species = "human";

// Attempt to reassign value
species = "werewolf";

console.log(species);
```

```
Output
Uncaught TypeError: Assignment to constant variable.
```

Since `const` values cannot be reassigned, they need to be declared and initialized at the same time, or will also throw an error.

```
// Declare but do not initialize a const
const todo;

console.log(todo);
```

```
Output
Uncaught SyntaxError: Missing initializer in const declaration
```

Values that cannot change in programming are known as *immutable*, while values that can be changed are *mutable*. Although `const` values cannot be reassigned, they are mutable as it is possible to modify the properties of objects declared with `const`.

```
// Create a car object with two properties
const car = {
    color: "blue",
    price: 15000
}

// Modify a property of car
car.price = 20000;

console.log(car);
```

```
Output
{ color: 'blue', price: 20000 }
```

Constants are useful for making it clear to your future self and other programmers working on a project with you that the intended variable should not be reassigned. If you expect that a variable may be modified in the future, you will likely want to use `let` to declare the variable instead.

Tags: variables, javascript