

# JavaScript Basics - Test

---

Name: \_

Termin: 6

Datum: 09.03.2017

## Data types and variables


1. Welche Datentypen sind primitive (skalare) Datentypen in JavaScript?

- Der `Date`-Datentyp.
- Der `number`-Datentyp.
- Der `boolean`-Datentyp.
- Der `function`-Datentyp.
- Der `string`-Datentyp.
- Der `Array`-Datentyp.

2. Welche Aussage ist korrekt?

- JavaScript arbeitet mit einem *block scope* Wertebereich und ist eine synchrone Sprache.
- JavaScript arbeitet mit einem *function scope* Wertebereich und ist eine synchrone Sprache.
- JavaScript arbeitet mit einem *block scope* Wertebereich und ist eine asynchrone Sprache.
- JavaScript arbeitet mit einem *function scope* Wertebereich und ist eine asynchrone Sprache.

3. Welchen Wert hat `result`?

```
 var given;  
  
given = 1;  
function changeGiven() {  
  given = 2;  
}  
given = 3;  
result = given;  
changeGiven();
```

- Es hat den Wert `result === undefined`.
- Es hat den Wert `result === 1`.

- Es hat den Wert `result === 2`.
- Es hat den Wert `result === 3`.

4. Welchen Wert hat `result`?

```
var given;

given = 1;
function changeGiven() {
  given = 2;
}
given = 3;
changeGiven();
result = given;
```

- Es hat den Wert `result === undefined`.
- Es hat den Wert `result === 1`.
- Es hat den Wert `result === 2`.
- Es hat den Wert `result === 3`.

5. Welchen Wert hat `result`?

```
var given;

function changeGiven() {
  given = 2;
}

result = given;
given = 3;
changeGiven();
```

- Es hat den Wert `result === undefined`.
- Es hat den Wert `result === 1`.
- Es hat den Wert `result === 2`.
- Es hat den Wert `result === 3`.

## Functions


1. Welche Funktionen werden gehoisted?

- Funktionen definiert mit `var myFunc = function() { /*...*/ };`.
- Funktionen definiert mit `function() { /*...*/ }`.
- Funktionen definiert mit `var myFunc = function myFunc() { /*...*/ };`.
- Funktionen definiert mit `function myFunc() { /*...*/ }`.

2. Welche Aussage ist korrekt?

- Eine Funktion ohne `return` gibt die zuletzt genutzte Variable zurück.
- Eine Funktion ohne `return` gibt `undefined` zurück.
- Eine Funktion ohne `return` gibt `null` zurück.
- Eine Funktion ohne `return` gibt nichts zurück.

3. Welchen Wert hat `result`?

```
 var obj;
```

```
  obj = {  
    value: 3,  
    increase: function() {  
      this.value += 1;  
      return this;  
    }  
  };  
};
```

```
var result = obj  
  .increase()  
  .increase()  
  .increase()  
  .value;
```

- Es hat den Wert `result === undefined`.
- Es hat den Wert `result === 3`.
- Es hat den Wert `result === 5`.
- Es hat den Wert `result === 6`.