CoreJS Interview1 Cheat Sheet

by itsMe via cheatography.com/189476/cs/39515/

JS Basics

Falsy ⇒ '', 0, null, undefined, NaN, fals

var - hoisting, let, const - blocks

Temporal Dead Zone начинается в начале области видимости переменной и заканчивается ее объявлением

Primitives: `Undefined, Boolean, String, Symbol, BigI nt, Null`

Non Primotives:
Object, Array, Function, RegExp, Date, Map, Set,
Error, Promise, enum

Strings

`text.l ength`

`text.s lic e(s tart, end)` => new string [start, end)

`text.s ubs tri ng( start, end)` new string [start, end)

`text.s ubs tr( start, length)` new string [start... `length` number of characters.

```
text.r epl ace (ol dValue, newValue) te
xt.r epl ace All (ol dValue, newValue)
text.t oUp per Case() text.t oLo wer Ca
se()
concat (st ring1, string2, ..., string
N)
text.t rim() trimSt art() trimEnd()
padSta rt( tar get Length, padString);
```

adds padding to the beginning

`padEnd (ta rge tLe ngth, padString)`:add s padding to the end

```
charAt (index) charCo deA t(i ndex)
```

`split( sep arator, limit)`

Timers

Методы setInterval(func, delay, ...args) и setTimeout(func, delay, ...args) позволяют выполнять func регулярно или только один раз после задержки delay, заданной в мс.

ля отмены выполнения необходимо вызвать clearInterval/clearTimeout со значением, которое возвращают методы setInterval/setTimeout.

Timers (cont)

Планирование с нулевой задержкой setTimeout(func, 0) — это же самое, setTimeout(func) используется для вызовов, которые должны быть исполнены как можно скорее, после завершения исполнения текущего кода.

Браузер ограничивает 4-мя мс минимальную задержку между вызовами, начиная с 5-го вызова.

```
let timerId = setTim eou t(...); clearT ime out (ti m
erId);
```

Objects

Creating Objects

Literal/ Contructor Function/ Class

```
{x: " y"} (new) functi on(){} new Object(
) same as {} class Banana {} Object.cr ea
te(proto[, proper tie sOb ject])
```

`delete book.u ser sRe ading`

Обход объекта

`for (let key in myObject)` `Object.en t`

`s() =>` `[key, value]`

`Object.keys()` ⇒ `[key]` **`Object.va lues()`** `=> [va lues]`

Методы

```
someOb j.h asO wnP roperty('key1')
obj.__proto__ === person ⇒ Object.ge tPr -
oto typ eOf (ob ject1) Object.se tPr oto
 typ eOf (ob ject1)
```

The **`valueOf()`** method of `[Object]` instances converts the `this` value [to an object]

Object.create()

```
function MyClass() {
    Sup erC las s.c all (this);
    Oth erS upe rCl ass.ca ll( -
this);
}
MyClas s.p rot otype =
Object.cr eat e(S upe rCl -
ass.pr oto type);
```

позволяет создавать новый объект и устанавливать прототип этого объекта в объект, переданный в качестве параметра

Events

Events имеют три фазы жизненного цикла: захват (capturing), цель (target) и всплытие (bubbling).

Разница между этими фазами заключается в порядке вызова обработчиков событий. В фазе захвата обработчики вызываются сверху вниз, а в фазе всплытия - снизу вверх. В фазе цели вызывается только один обработчик, связанный с элементом цели.

Mouse Events: - click - dblclick - mousedown - mouseup - mousemove - mouseover - mouseout - contextmenu

Keyboard Events: - keydown - keyup - keypress

Form: formdata reset submit;

Input:change, input, cut, copy, paste

Array NOT Mutating

```
concat(array1, array2, ..., arrayN
array.f il ter (ca llb ack(el, ind
Array
array.m ap (ca llb ack (elt, ind,
ray
array.s li ce( start, end): new Ar
array.r ed uce (ca llb ack (acc, c
ny
reduce Rig ht( cal lba ck(acc, cur
```

Array Mutations

```
push(e lem ent1, element2, ..., el
pop()
shift()
unshif t(e lem ent1, element2, ...
splice (start, delete Count, item1
N)
sort(c omp are Fun ction)
reverse()
fill(v alue, start, end)
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com

Iterate Array

```
for (let i = 0; i <
scores.length; i++) {}
for (i in scores) {}
for (score of scores) {}
scores.fo rEa ch( (score) =>
{});
every( fun ction (element,
index, array) { / … / },
thisArg)
array.r educe( functi on( total,
el, i, arr),i nit ial Value )
every( fun ction (element,
index, array) { / … / },
thisArg)
```

Copy Array

```
const arrNew = [...arrOld];
const newArray = [...ol dAr -
ray];
{ const json = JSON.s tri ngi -
fy( old Array);
const newArray = JSON.p ars -
e(j son); }
const arrNew = arrOld.sl ice();
const arrNew = Array.f ro m(a -
rrOld);
const arrNew = [].con cat (ar -
rOld);
```


Sponsored by **CrosswordCheats.com**
Learn to solve cryptic crosswords!
http://crosswordcheats.com