DATE OBJECT & CLASS

Presented by Azizbayli Cavad

DATE OBJECT

Date obyekti tarix və vaxtlar (saatlar) ilə işləmək üçün istifadə olunmaqdadır.

Date obyektlərini new Date() vasitəsilə yaradırıq

Default olaraq JS browser-ın time zone-u işlədərək nəticəni text string olaraq çıxaracaq

Tarixi əldə etmənin 4 yolu var:

var d = new Date();

var d = new Date(*milliseconds*);

var d = new Date(*dateString*);

var d = new Date(*year*, *month*, *day*, *hours*, *minutes*, *seconds*, *milliseconds*);

New Date() hal-hazırkı tarix və vaxtı göstərən yeni bir obyekt yaradır. (obyektlər statikdir. Yəni kompyuterdə zaman davam etsə də obyektlərdə dəyişmir)

DATE OBJECT

Qeyd: JS ayları 0-11 aralığında sayır. Yəni **January = 0. December = 11.**

Overflow – lar , yəni bu aralıqdan kənara çıxmalar aylarda növbəti ilə, günlərdə isə növbəti aya əlavə olunur.

```
const d = new Date(2018, 11, 24, 10, 33, 30);
```

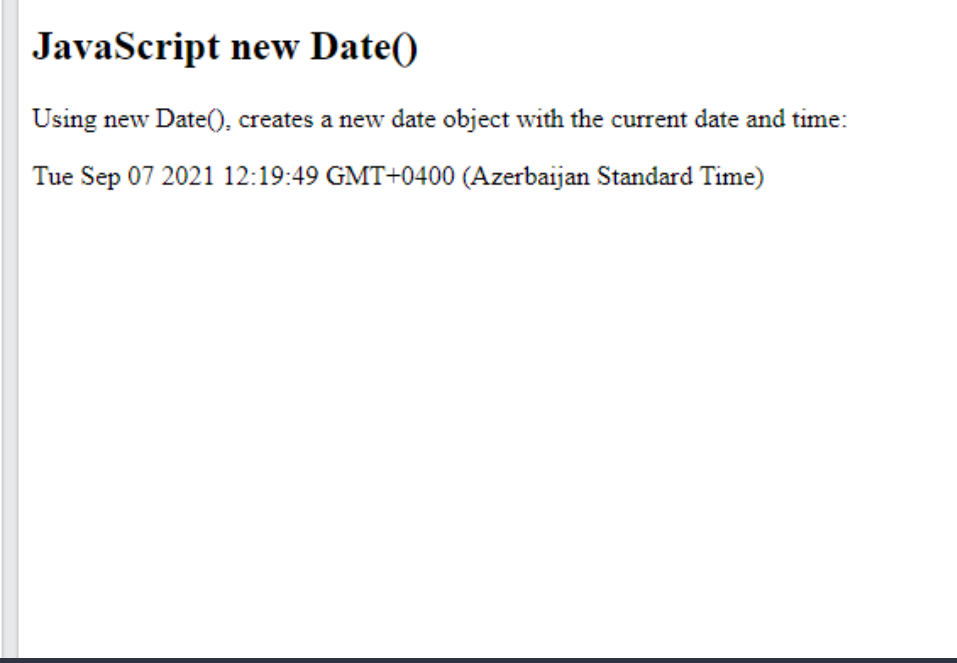
Burdakı 6 rəqəm uyğun olaraq il, ay, gün, saat, dəqiqə, saniyə -i təmsil edir.

JS tarixləri millisaniyə olaraq yadda saxlayır.

const d = new Date(0); əmri zero time üzərinə verilmiş millisaniyəni əlavə edərək yeni obyekt yaradır.

JS-də zero time “January 01, 1970 00:00:00 UTC”-dır

```html
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript new Date()</h2>
<p>Using new Date(), creates a new date object with the current date and time:</p>

<p id="demo"></p>

<script>
const d = new Date();
document.getElementById("demo").innerHTML = d;
</script>

</body>
</html>
```



DATE METHODS

- Date obyekti yaratdıqda bir sıra metodlar onun üzərində əməliyyat icra etməmizə imkan yaradır.
- Bu metodlar bizə ili, günü, saatı, dəqiqəni, saniyəni və millisaniyəni get və set etməmizə icazər verir.

Method	Description
getDate()	Returns the day of the month (from 1-31)
getDay()	Returns the day of the week (from 0-6)
getFullYear()	Returns the year
getHours()	Returns the hour (from 0-23)
getMilliseconds()	Returns the milliseconds (from 0-999)
getMinutes()	Returns the minutes (from 0-59)
getMonth()	Returns the month (from 0-11)
getSeconds()	Returns the seconds (from 0-59)
getTime()	Returns the number of milliseconds since midnight Jan 1 1970, and a specified date
getTimezoneOffset()	Returns the time difference between UTC time and local time, in minutes
getUTCDate()	Returns the day of the month, according to universal time (from 1-31)
getUTCDay()	Returns the day of the week, according to universal time (from 0-6)
getUTCFullYear()	Returns the year, according to universal time
getUTCHours()	Returns the hour, according to universal time (from 0-23)
getUTCMilliseconds()	Returns the milliseconds, according to universal time (from 0-999)
getUTCMinutes()	Returns the minutes, according to universal time (from 0-59)
getUTCMonth()	Returns the month, according to universal time (from 0-11)
getUTCSeconds()	Returns the seconds, according to universal time (from 0-59)
getYear()	Deprecated. Use the getFullYear() method instead
now()	Returns the number of milliseconds since midnight Jan 1, 1970
parse()	Parses a date string and returns the number of milliseconds since January 1, 1970
parse()	Parses a date string and returns the number of milliseconds since January 1, 1970
setDate()	Sets the day of the month of a date object
setFullYear()	Sets the year of a date object
setHours()	Sets the hour of a date object
setMilliseconds()	Sets the milliseconds of a date object
setMinutes()	Set the minutes of a date object
setMonth()	Sets the month of a date object
setSeconds()	Sets the seconds of a date object
setTime()	Sets a date to a specified number of milliseconds after/before January 1, 1970
setUTCDate()	Sets the day of the month of a date object, according to universal time
setUTCFullYear()	Sets the year of a date object, according to universal time
setUTCHours()	Sets the hour of a date object, according to universal time
setUTCMilliseconds()	Sets the milliseconds of a date object, according to universal time
setUTCMinutes()	Set the minutes of a date object, according to universal time
setUTCMonth()	Sets the month of a date object, according to universal time
setUTCSeconds()	Set the seconds of a date object, according to universal time
setYear()	Deprecated. Use the setFullYear() method instead
toDateString()	Converts the date portion of a Date object into a readable string
toGMTString()	Deprecated. Use the toUTCString() method instead
toISOString()	Returns the date as a string, using the ISO standard
toJSON()	Returns the date as a string, formatted as a JSON date
toLocaleDateString()	Returns the date portion of a Date object as a string, using locale conventions

CLASS

Class bir növ funksiyadır, ancaq function keyword-ü əvəzinə biz class keywod-ü istifadə edirik. Və properti-lər constructor()-ın içərisində təyin olunur.

Class-ı class açarsözü vasitəsilə yaradırıq və həmisə constructor() metodu əlavə edirik. Class obyekt deyil, obyektlər üçün template-dir.

Class-ı obyekt yaratmaq üçün istifadə edə bilərik.

```html
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Class</h2>

<p>In this example we demonstrate a simple class definition and how to use it.</p>

<p id="demo"></p>

<script>
class Car {
  constructor(brand) {
    this.carname = brand;
  }
}

mycar = new Car("Ford");

document.getElementById("demo").innerHTML = mycar.carname;
</script>

</body>
</html>
```

JavaScript Class

In this example we demonstrate a simple class definition and how to use it.

Ford

CONSTRUCTOR & METHODS

Constructor metodu xüsusi bir metoddur.

Onun adı mütləq olaraq “constructor” olmalıdır

Yeni bir obyekt yaradıldıqda o avtomatik olaraq execute olunur

Obyekt propertilərini initialize etmək üçün istifadə olunur

Class metodları obyektlərdəki kimi yaradılır. Class-ı yaradırıq sonra constructor metodunu əlavə edirik daha sonra isə istədiyimiz metodları.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Class Method</h2>

<p>How to define and use a Class method.</p>

<p id="demo"></p>

<script>
class Car {
  constructor(name, year) {
    this.name = name;
    this.year = year;

  }
  age() {
    let date = new Date();
    return date.getFullYear() - this.year;
  }
}

let myCar = new Car("Ford", 2014);
document.getElementById("demo").innerHTML =
"My car is " + myCar.age() + " years old.";
</script>

</body>
</html>
```

JavaScript Class Method

How to define and use a Class method.

My car is 7 years old.

Extends açar söze başqa bir class-ın child class-ı yaratmaq üçün istifadə olunur. Child class bütün metodları valideyn class-dan miras alır.

Static açar sözü class-lar üçün static metodlar təyin etmək üçün istifadə olunur. Static metodlar birbaşa class üzərindən çağırılır, obyekt yaratmadan.

Super() metodu vasitəsilə biz valideyn class-ın constructor metodunu çağıra və onu properti və metodlarına çata bilirik

Class Keywords

Keyword	Description
extends	Extends a class (inherit)
static	Defines a static method for a class
super	Refers to the parent class

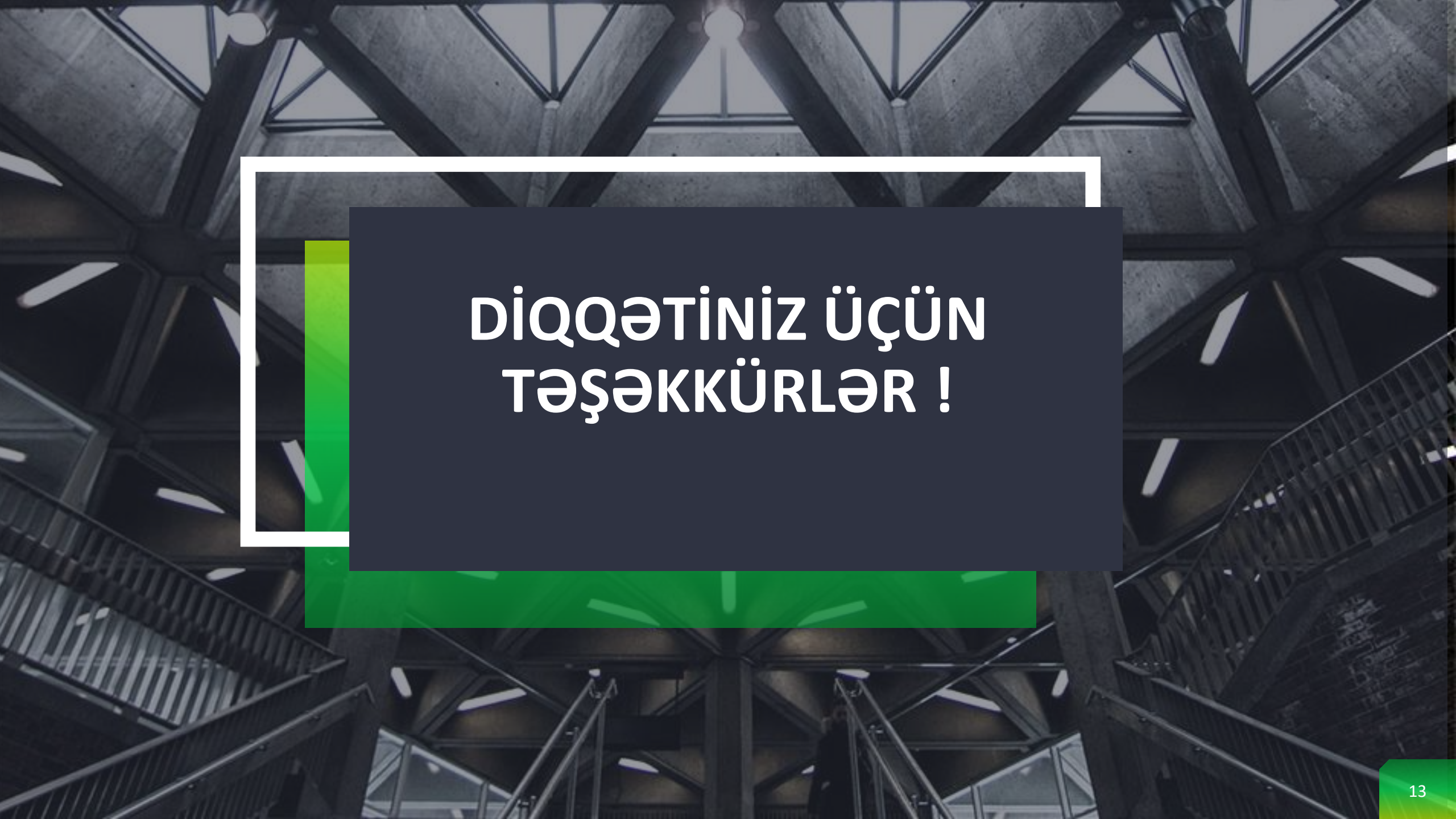
```html
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Class Inheritance</h2>

<p>Use the "extends" keyword to inherit all methods from another class.</p>
<p>Use the "super" method to call the parent's constructor function.</p>

<p id="demo"></p>

<script>
class Car {
  constructor(brand) {
    this.carname = brand;
  }
  present() {
    return 'I have a ' + this.carname;
  }
}

class Model extends Car {
  constructor(brand, mod) {
    super(brand);
    this.model = mod;
  }
  show() {
    return this.present() + ', it is a ' + this.model;
  }
}

mycar = new Model("Ford", "Mustang");
document.getElementById("demo").innerHTML = mycar.show();
</script>

</body>
</html>
```

JavaScript Class Inheritance

Use the "extends" keyword to inherit all methods from another class.

Use the "super" method to call the parent's constructor function.

I have a Ford, it is a Mustang



DİQQƏTİNİZ ÜÇÜN TƏŞƏKKÜRLƏR !