Дата и время: встроенный объект Date

Объект **Date** предназначен для манипуляций с датами и временами. Его примитивным значением является число, равное количеству миллисекунд относительно *базового времени*, равного полуночи 1 января 1970 г. по Гринвичскому меридиану (UTC, Universal Coordinated Time). Если это значение равно **NaN**, то оно считается неопределенным.

День состоит из 86400000 миллисекунд. Диапазон значений Date от -100000000 дней до 100000000 дней относительно базового времени, что приблизительно равно 285616 лет в каждую сторону отсчета.

Для создания объектов Date используются следующие конструкторы:

```
new Date()
new Date(число)
new Date(строка)
new Date(год, месяц, день [,часы [,минуты [,секунды [,мс]?]?]?]?)
```

Здесь:

- *число* — числовое выражение, задающее примитивное значение объекта в миллисекундах;
- *строка* — строковое выражение, задающее дату и время в формате, описанном в методе **parse**;
- *год* — числовое выражение, задающее полный номер года (например, 1988, а не 88);
- *месяц* — числовое выражение, задающее номер месяца (0 = январь, 1 = февраль, …, 11 = декабрь);
- *день* — числовое выражение, задающее номер для в месяце от 1 до 31;
- *часы* — необязательное числовое выражение, задающее номер часа от 0 до 23;
- *минуты* — необязательное числовое выражение, задающее номер минуты от 0 до 59;
- *секунды* — необязательное числовое выражение, задающее номер секунды от 0 до 59;
- *мс* — необязательное числовое выражение, задающее номер миллисекунды от 0 до 999.

Первый конструктор создает объект **Date** с текущей датой и временем по местному времени. Остальные конструкторы создают объект **Date** с датой и временем, заданными аргументами конструктора.

Примеры:

```
var a = new Date();                       // текущие дата и время
var b = new Date("May 21, 1958 10:15 AM"); // заданные дата и время
var c = new Date(1958, 4, 21, 10, 15);     // то же самое в другом формате
```

Свойства объекта Date

Свойство	Описание	Член прототипа
constructor	Конструктор, который создал объект.	Да
prototype	Ссылка на прототип класса объектов.	Нет

Методы объекта Date

Метод	Описание	Член прототипа
getDate	Возвращает день месяца по местному времени.	Да
getDay	Возвращает день недели по местному времени.	Да
getFullYear	Возвращает полный номер года по местному времени.	Да
getHours	Возвращает часы по местному времени.	Да
getMilliseconds	Возвращает миллисекунды по местному времени.	Да
getMinutes	Возвращает минуты по местному времени.	Да
getMonth	Возвращает месяц по местному времени.	Да
getSeconds	Возвращает секунды по местному времени.	Да
getTime	Возвращает примитивное значение объекта.	Да
getTimezoneOffset	Возвращает разницу в минутах между местным временем и UTC.	Да
getUTCDate	Возвращает день месяца по времени UTC.	Да
getUTCDay	Возвращает день недели по времени UTC.	Да
getUTCFullYear	Возвращает полный номер года по времени UTC.	Да
getUTCHours	Возвращает часы по времени UTC.	Да
getUTCMilliseconds	Возвращает миллисекунды по времени UTC.	Да
getUTCMinutes	Возвращает минуты по времени UTC.	Да
getUTCMonth	Возвращает месяц по времени UTC.	Да
getUTCSeconds	Возвращает секунды по времени UTC.	Да
getVarDate	Возвращает примитивное значение объекта в формате VT_DATE.	Нет
getYear	Возвращает номер года по местному времени.	Да
parse	Преобразует дату, заданную строкой, в количество миллисекунд относительно базового времени.	Нет
setDate	Устанавливает день месяца по местному времени.	Да

setFullYear	Устанавливает полный номер года по местному времени.	Да
setHours	Устанавливает часы по местному времени.	Да
setMilliseconds	Устанавливает миллисекунды по местному времени.	Да
setMinutes	Устанавливает минуты по местному времени.	Да
setMonth	Устанавливает месяц по местному времени.	Да
setSeconds	Устанавливает секунды по местному времени.	Да
setTime	Устанавливает примитивное значение объекта.	Да
setUTCDate	Устанавливает день месяца по времени UTC.	Да
setUTCFullYear	Устанавливает полный номер года по времени UTC.	Да
setUTCHours	Устанавливает часы по времени UTC.	Да
setUTCMilliseconds	Устанавливает миллисекунды по времени UTC.	Да
setUTCMinutes	Устанавливает минуты по времени UTC.	Да
setUTCMonth	Устанавливает месяц по времени UTC.	Да
setUTCSeconds	Устанавливает секунды по времени UTC.	Да
setYear	Устанавливает номер года по местному времени.	Да
toDateString	Преобразует примитивное значение объекта в строку даты.	Да
toGMTString	Преобразует примитивное значение объекта в строку даты и времени по Гринвичскому меридиану.	Да
toLocaleDateString	Преобразует примитивное значение объекта в строку даты в формате операционной системы.	Да
toLocaleString	Преобразует примитивное значение объекта в строку даты и времени в формате операционной системы.	Да
toLocaleTimeString	Преобразует примитивное значение объекта в строку времени в формате операционной системы.	Да
toString	Преобразует примитивное значение объекта в строку.	Да
toTimeString	Преобразует примитивное значение объекта в строку времени.	Да
toUTCString	Преобразует примитивное значение объекта в строку даты и времени по UTC.	Да
UTC	Возвращает примитивное значение объекта по времени UTC.	Нет

valueOf	Возвращает примитивное значение объекта.	Да

Метод getDate

Синтаксис: *дата*.getDate()
Результат: числовое значение

Метод **getDate** преобразует примитивное значение объекта *дата* в номер дня месяца по местному времени и возвращает его. Результатом является целое число в диапазоне от 1 до 31. Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getDate());
```

выведет на экран обозревателя число 25.

Метод getDay

Синтаксис: *дата*.getDay()
Результат: числовое значение

Метод **getDay** преобразует примитивное значение объекта *дата* в номер дня недели по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 6:

0 = воскресенье, 1 = понедельник, 2 = вторник, 3 = среда, 4 = четверг, 5 = пятница, 6 = суббота.

Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getDay());
```

выведет на экран обозревателя число 6 (т. е. рождество в 1999 г. пришлось на субботу).

Метод getFullYear

Синтаксис: *дата*.getFullYear()
Результат: числовое значение

Метод **getFullYear** преобразует примитивное значение объекта *дата* в полный (четырехзначный) номер года по местному времени и возвращает его. Например, сценарий

```
var today = new Date();
document.write(today.getFullYear());
```

выведет на экран обозревателя число 2000.

Метод getHours

Синтаксис: *дата*.getHours()
Результат: числовое значение

Метод **getHours** преобразует примитивное значение объекта *дата* в номер часа по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 23. Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getHours());
```

выведет на экран обозревателя число 23.

Метод getMilliseconds

Синтаксис: *дата*.getMilliseconds()
Результат: числовое значение

Метод **getMilliseconds** преобразует примитивное значение объекта *дата* в микросекунды по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 999. Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getMilliseconds());
```

выведет на экран обозревателя число 0.

Метод getMinutes

Синтаксис: *дата*.getMinutes()
Результат: числовое значение

Метод **getMinutes** преобразует примитивное значение объекта *дата* в минуты по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 59. Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getMinutes());
```

выведет на экран обозревателя число 59.

Метод getMonth

Синтаксис: *дата*.getMonth()
Результат: числовое значение

Метод **getMonth** преобразует примитивное значение объекта *дата* в номер месяца по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 11: 0 = январь, 1 = февраль, 2 = март, 3 = апрель, 4 = май, 5 = июнь, 6 = июль, 7 = август, 8 = сентябрь, 9 = октябрь, 10 = ноябрь, 11 = декабрь.

Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getMonth());
```

выведет на экран обозревателя число `11`.

Метод getSeconds

Синтаксис: *дата*.getSeconds()
Результат: числовое значение

Метод **getSeconds** преобразует примитивное значение объекта *дата* в секунды по местному времени и возвращает его. Результатом является целое число в диапазоне от 0 до 59. Например, сценарий

```
var xmas = new Date("December 25, 1999 23:59:59");
document.write(xmas.getSeconds());
```

выведет на экран обозревателя число `59`.

Метод getTime

Синтаксис: *дата*.getTime()
Результат: числовое значение

Метод **getTime** возвращает примитивное значение объекта *дата*. Этот метод полезен для копирования значения одного объекта **Date** в другой. Пример:

```
var d1 = new Date("May 21, 1958");
var d2 = new Date();
d2.SetTime(d1.GetTime());      // теперь значения этих объектов равны
```

Метод getTimezoneOffset

Синтаксис: *дата*.getTimezoneOffset()
Результат: числовое значение

Метод **getTimezoneOffset** возвращает разницу в минутах между временем UTC и местным временем (т. е. временем компьютера, на котором выполняется сценарий). Из-за переходов на летнее и зимнее время эта разница в течение года может изменяться. Для перевода местного времени в UTC следует *прибавить* к нему полученное значение.

Если, например, следующий сценарий

```
var today = new Date();
document.write(today.getTimezoneOffset() / 60);
```

выполняется на компьютере, находящемся в Екатеринбурге, то он выведет на экран обозревателя число `-6`. Это означает, что время в Екатеринбурге на 6 часов *больше*, чем UTC.

Метод getUTCDate

Синтаксис: *дата*.getUTCDate()
Результат: числовое значение

Метод **getUTCDate** преобразует примитивное значение объекта *дата* в номер дня месяца по времени UTC и возвращает его. Результатом является целое число в диапазоне от 1 до 31. Например, сценарий

```
var today = new Date();
document.write(today.getUTCDate());
```

выведет на экран обозревателя сегодняшний день месяца.

Метод getUTCDay

Синтаксис: *дата*.getUTCDay()
Результат: числовое значение

Метод **getUTCDay** преобразует примитивное значение объекта *дата* в номер дня недели по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 6:

0 = воскресенье, 1 = понедельник, 2 = вторник, 3 = среда, 4 = четверг, 5 = пятница, 6 = суббота.

Например, сценарий

```
var today = new Date();
document.write(today.getUTCDay());
```

выведет на экран обозревателя сегодняшний день недели.

Метод getUTCFullYear

Синтаксис: *дата*.getUTCFullYear()
Результат: числовое значение

Метод **getUTCFullYear** преобразует примитивное значение объекта *дата* в полный (четырехзначный) номер года по времени UTC и возвращает его. Например, сценарий

```
var today = new Date();
document.write(today.getUTCFullYear());
```

выведет на экран обозревателя номер текущего года.

Метод getUTCHours

Синтаксис: *дата*.getHours()
Результат: числовое значение

Метод **getUTCHours** преобразует примитивное значение объекта *дата* в номер часа по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 23. Например, сценарий

```
var today = new Date();
document.write(today.getUTCHours());
```

выведет на экран обозревателя текущее количество часов.

Метод getUTCMilliseconds

Синтаксис: *дата*.getUTCMilliseconds()
Результат: числовое значение

Метод **getUTCMilliseconds** преобразует примитивное значение объекта *дата* в микросекунды по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 999. Например, сценарий

```
var today = new Date();
document.write(today.getUTCMilliseconds());
```

выведет на экран обозревателя текущее количество миллисекунд.

Метод getUTCMinutes

Синтаксис: *дата*.getUTCMinutes()
Результат: числовое значение

Метод **getUTCMinutes** преобразует примитивное значение объекта *дата* в минуты по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 59. Например, сценарий

```
var today = new Date();
document.write(today.getUTCMinutes());
```

выведет на экран обозревателя текущее количество минут.

Метод getUTCMonth

Синтаксис: *дата*.getUTCMonth()
Результат: числовое значение

Метод **getUTCMonth** преобразует примитивное значение объекта *дата* в номер месяца по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 11: 0 = январь, 1 = февраль, 2 = март, 3 = апрель, 4 = май, 5 = июнь, 6 = июль, 7 = август, 8 = сентябрь, 9 = октябрь, 10 = ноябрь, 11 = декабрь.

Например, сценарий

```
var today = new Date();
document.write(today.getUTCMonth());
```

выведет на экран обозревателя номер текущего месяца.

Метод getUTCSeconds

```
Синтаксис: дата.getSeconds()
Результат: числовое значение
```

Метод **getUTCSeconds** преобразует примитивное значение объекта *дата* в секунды по времени UTC и возвращает его. Результатом является целое число в диапазоне от 0 до 59. Например, сценарий

```
var today = new Date();
document.write(today.getUTCSeconds());
```

выведет на экран обозревателя текущее количество секунд.

Метод getVarDate

```
Синтаксис: дата.getVarDate()
Результат: числовое значение
```

Метод **getVarDate** преобразует примитивное значение объекта *дата* в формат VT_DATE и возвращает его. Результат зависит от региональных настроек и не используется JavaScript. Поэтому этот метод предназначен исключительно для взаимодействия с COM-объектами, объектами ActiveX или с языками, которые используют формат даты VT_DATE (таковы, например, Visual Basic и VBScript).

Метод getYear

```
Синтаксис: дата.getYear()
Результат: числовое значение
Поддержка: IE - Годы 1900-1999 возвращаются как 00-99; все остальные годы
              возвращаются четырехзначными числами.
           NN - Возвращается номер года минус 1900.
```

Метод **getYear** преобразует примитивное значение объекта *дата* в номер года по местному времени и возвращает его. Этот метод является устаревшим и поддерживается только в целях совместимости; вместо него следует пользоваться методом **getFullYear**.

Например, сценарий

```
var today = new Date();
document.write(today.getYear());
```

выведет на экран Internet Explorer число `2000`, а на экран Netscape Navigator число `100`.

Метод parse

```
Синтаксис: Date.parse(строка)
```

Аргументы: *строка* — строковое выражение
Результат: числовое значение

Метод **parse** преобразует дату, заданную *строкой*, в количество миллисекунд относительно базового времени, и возвращает его. *Строка*: должна содержать дату и время в определенном формате. Допускаются следующие форматы даты (пробелы и запятые трактуются как разделители):

```
"7/20/82"    "July 10, 1995"     "12-11-76"    "10 Jan 2000"
```

Время может содержать часы, минуты и секунды, разделенные двоеточием:

```
"11:"    "11:45"     "23:45:12"    "11:45:12 PM"
```

После даты и времени может стоять указатель временной зоны следующего вида (обе формы эквивалентны и означают на 4 часа 30 минут больше UTC):

```
"UTC+0430"    "GMT+0430"
```

Если такого указателя нет, то время считается местным. Перед датой и временем может стоять название дня недели; если оно не соответствует заданной дате, то игнорируется.

Это статический метод объекта **Date**, поэтому для доступа к нему используется форма `Date.parse`, а не *`объект`*`.parse`.

Например, сценарий

```
var d = new Date();
d.setTime(Date.parse("21 May 1958 10:12"));
document.write(d.toLocaleString());
```

выведет на экран обозревателя примерно такую строку: `Wednesday, May 21, 1958 10:12:00` (точный вид этой строки зависит от обозревателя и настроек операционной системы).

Метод setDate

Синтаксис: *дата*`.setDate(`*день*`)`
Аргументы: *день* — числовое выражение
Результат: нет

Метод **setDate** изменяет номер *дня* месяца по местному времени в объекте *дата*. *День* должен быть целым числом в диапазоне от 1 до 31. Если *день* превышает количество дней в данном месяце, то устанавливается день, равный *дню* минус количество дней в данном месяце, а номер месяца увеличивается. Если *день* отрицателен, то день и месяц изменяются по аналогичным правилам.

Например, сценарий

```
var d = new Date("July 27, 1962 23:30:00");
d.setDate(32);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Wednesday, August 01, 1962 23:30:00`.

Метод setFullYear

Синтаксис: *дата*.setFullYear(*год* [,*месяц* [,*день*]?]?)
Аргументы: *год*, *месяц*, *день* — числовые выражения
Результат: нет

Метод **setFullYear** изменяет полный (четырехзначный) номер *года* по местному времени в объекте *дата*. Одновременно он позволяет изменить номер *месяца* и номер *дня* в месяце (ср. методы **setMonth** и **setDate**). Например, сценарий

```
var d = new Date("September 26, 2000 09:52:30");
d.setFullYear(1999);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Sunday, September 26, 1999 09:52:30`.

Метод setHours

Синтаксис: *дата*.setHours(*часы* [,*минуты* [,*секунды* [,*миллисекунды*]?]?]?)
Аргументы: *часы*, *минуты*, *секунды*, *миллисекунды* — числовые выражения
Результат: нет

Метод **setHours** задает *часы* по местному времени в объекте *дата*. *Часы* должны быть целым числом в диапазоне от 0 до 23. Если *часы* больше 23, то устанавливается час, равный *часам* минус 23, а номер дня увеличивается. Если *часы* отрицательны, то час и день изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *минуты*, *секунды* и *миллисекунды* в объекте *дата*. Ср. методы **setMinutes**, **setSeconds** и **setMilliseconds**.

Например, сценарий

```
var d = new Date("July 27, 1962 23:30:00");
d.setHours(25);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Saturday, July 28, 1962 01:30:00`.

Метод setMilliseconds

Синтаксис: *дата*.setMilliseconds(*миллисекунды*)

Аргументы: *миллисекунды* — числовое выражение
Результат: нет

Метод **setMilliseconds** задает *миллисекунды* по местному времени в объекте *дата*. *Миллисекунды* должны быть целым числом в диапазоне от 0 до 999. Если *миллисекунды* больше 999, то устанавливаются миллисекунды, равные *миллисекундам* минус 999, а номер секунды увеличивается. Если *миллисекунды* отрицательны, то миллисекунды и секунды изменяются по аналогичным правилам.

Например, сценарий

```
var d = new Date("January 20, 1999 23:59:59");
d.setMilliseconds(-1000);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Wednesday, January 20, 1999 23:59:58`.

Метод setMinutes

Синтаксис: *дата*.setMinutes(*минуты* [,*секунды* [,*миллисекунды*]?]?)
Аргументы: *минуты*, *секунды*, *миллисекунды* — числовое выражение
Результат: нет

Метод **setMinutes** задает *минуты* по местному времени в объекте *дата*. *Минуты* должны быть целым числом в диапазоне от 0 до 59. Если *минуты* больше 59, то устанавливаются минуты, равные *минутам* минус 59, а номер часа увеличивается. Если *минуты* отрицательны, то минуты и часы изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *секунды* и *миллисекунды* в объекте *дата*. Ср. методы **setSeconds** и **setMilliseconds**.

Например, сценарий

```
var d = new Date("January 20, 1999 23:59:59");
d.setMinutes(60);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Thursday, January 21, 1999 00:00:59`.

Метод setMonth

Синтаксис: *дата*.setMonth(*месяц* [,день]?)
Аргументы: *месяц*, *день* — числовые выражения
Результат: нет

Метод **setMonth** задает номер *месяца* по местному времени в объекте *дата*. *Месяц* должен быть целым числом в диапазоне от 0 до 11.

Если *месяц* больше 11, то устанавливается месяц, равный *месяцу* минус 11, а номер года увеличивается. Если *месяц* отрицателен, то месяц и годы изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить номер *дня* в месяце (ср. метод **setDate**).

Например, сценарий

```
var d = new Date("January 20, 1999 23:59:59");
d.setMonth(-1);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Sunday, December 20, 1998 23:59:59`.

Метод setSeconds

Синтаксис: *дата*.setSeconds(*секунды* [,*миллисекунды*]?)
Аргументы: *секунды*, *миллисекунды* — числовые выражения
Результат: нет

Метод **setSeconds** задает *секунды* по местному времени в объекте *дата*. *Секунды* должны быть целым числом в диапазоне от 0 до 59. Если *секунды* больше 59, то устанавливаются секунды, равные *секундам* минус 59, а номер минуты увеличивается. Если *секунды* отрицательны, то секунды и минуты изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *микросекунды* (ср. метод **setMilliseconds**).

Например, сценарий

```
var d = new Date("January 20, 1999 23:59:59");
d.setSeconds(60);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Thursday, January 21, 1999 00:00:00`.

Метод setTime

Синтаксис: *дата*.setTime(*значение*)
Аргументы: *значение* — числовое выражение
Результат: числовое значение

Метод **setTime** устанавливает примитивное значение объекта *дата*, равным *значению*. Заданное *значение* объекта не зависит от временной зоны. Этот метод полезен для копирования значения одного объекта **Date** в другой. Пример:

```
var d1 = new Date("May 21, 1958");
var d2 = new Date();
d2.SetTime(d1.GetTime());      // теперь значения этих объектов равны
```

Метод setUTCDate

Синтаксис: *дата*.setUTCDate(*день*)
Аргументы: *день* — числовое выражение
Результат: нет

Метод **setUTCDate** изменяет номер *дня* месяца по времени UTC в объекте *дата*. *День* должен быть целым числом в диапазоне от 1 до 31. Если *день* превышает количество дней в данном месяце, то устанавливается день, равный *дню* минус количество дней в данном месяце, а номер месяца увеличивается. Если *день* отрицателен, то день и месяц изменяются по аналогичным правилам. Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCDate(25);
```

Метод setUTCFullYear

Синтаксис: *дата*.setUTCFullYear(*год* [,*месяц* [,*день*]?]?)
Аргументы: *год*, *месяц*, *день* — числовые выражения
Результат: нет

Метод **setUTCFullYear** изменяет полный (четырехзначный) номер *года* по времени UTC в объекте *дата*. Одновременно он позволяет изменить номер *месяца* и номер *дня* в месяце (ср. методы **setUTCMonth** и **setUTCDate**). Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCFullYear(1978);
```

Метод setUTCHours

Синтаксис: *дата*.setUTCHours(*часы* [,*минуты* [,*секунды* [,*миллисекунды*]?]?]?)
Аргументы: *часы*, *минуты*, *секунды*, *миллисекунды* — числовые выражения
Результат: нет

Метод **setUTCHours** задает *часы* по времени UTC в объекте *дата*. *Часы* должны быть целым числом в диапазоне от 0 до 23 . Если *часы* больше 23, то устанавливается час, равный *часам* минус 23, а номер дня увеличивается. Если *часы* отрицательны, то час и день изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *минуты*, *секунды* и *миллисекунды* в объекте *дата*. Ср. методы **setUTCMinutes**, **setUTCSeconds** и **setUTCMilliseconds**. Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCHours(22);
```

Метод setUTCMilliseconds

Синтаксис: *дата*.setUTCMilliseconds(*миллисекунды*)
Аргументы: *миллисекунды* — числовое выражение
Результат: нет

Метод **setUTCMilliseconds** задает *миллисекунды* по времени UTC в объекте *дата*. *Миллисекунды* должны быть целым числом в диапазоне от 0 до 999. Если *миллисекунды* больше 999, то устанавливаются миллисекунды, равные *миллисекундам* минус 999, а номер секунды увеличивается. Если *миллисекунды* отрицательны, то миллисекунды и секунды изменяются по аналогичным правилам. Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCMilliseconds(999);
```

Метод setUTCMinutes

Синтаксис: *дата*.setUTCMinutes(*минуты* [,*секунды* [,*миллисекунды*]?]?)
Аргументы: *минуты*, *секунды*, *миллисекунды* — числовое выражение
Результат: нет

Метод **setUTCMinutes** задает *минуты* по времени UTC в объекте *дата*. *Минуты* должны быть целым числом в диапазоне от 0 до 59. Если *минуты* больше 59, то устанавливаются минуты, равные *минутам* минус 59, а номер часа увеличивается. Если *минуты* отрицательны, то минуты и часы изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *секунды* и *миллисекунды* в объекте *дата*. Ср. методы **setUTCSeconds** и **setUTCMilliseconds**. Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCMinutes(31);
```

Метод setUTCMonth

Синтаксис: *дата*.setUTCMonth(*месяц* [,день]?)
Аргументы: *месяц*, *день* — числовые выражения
Результат: нет

Метод **setUTCMonth** задает номер *месяца* по времени UTC в объекте *дата*. *Месяц* должен быть целым числом в диапазоне от 0 до 11. Если *месяц* больше 11, то устанавливается месяц, равный *месяцу* минус 11, а номер года увеличивается. Если *месяц* отрицателен, то месяц и годы изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить номер *дня* в месяце (ср. метод **setUTCDate**). Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCMonth(11);
```

Метод setUTCSeconds

Синтаксис: *дата*.setUTCSeconds(*секунды* [,*миллисекунды*]?)
Аргументы: *секунды*, *миллисекунды* — числовые выражения
Результат: нет

Метод **setUTCSeconds** задает *секунды* по времени UTC в объекте *дата*. *Секунды* должны быть целым числом в диапазоне от 0 до 59. Если *секунды* больше 59, то устанавливаются секунды, равные *секундам* минус 59, а номер минуты увеличивается. Если *секунды* отрицательны, то секунды и минуты изменяются по аналогичным правилам.

Этот метод позволяет одновременно изменить *микросекунды* (ср. метод **setUTCMilliseconds**). Пример:

```
var d = new Date("July 27, 1962 23:30:00");
d.setUTCSeconds(59);
```

Метод setYear

Синтаксис: *дата*.setYear(*год*)
Аргументы: *год* — числовое выражение
Результат: нет

Метод **setYear** устанавливает номер года по местному времени в объекте *дата*. Если *год* лежит в диапазоне от 0 до 99 включительно, то устанавливается номер года, равный 1900 + *год*, в остальных случаях устанавливается номер года, равный *году*. Этот метод является устаревшим и поддерживается только в целях совместимости; вместо него следует пользоваться методом **setFullYear**.

Например, сценарий

```
var d = new Date("May 1, 2000 20:00:00");
d.setYear(99);
document.write(d.toLocaleString());
```

выведет на экран обозревателя строку `Saturday, May 01, 1999 20:00:00`.

Метод toDateString

Синтаксис: *дата*.toDateString()
Результат: строковое выражение
Поддержка: IE - Поддерживается с версии 5.5.
NN - Не поддерживается.

Метод **toDateString** преобразует примитивное значение объекта *дата* в строку даты по местному времени. Например, сценарий

```
var d = new Date();
document.write(d.toDateString());
```

выведет на экран обозревателя Internet Explorer строку вида `Sun Oct 29 2000`.

Метод toGMTString

Синтаксис: *дата*.toGMTString()
Результат: строковое выражение

Метод **toGMTString** преобразует примитивное значение объекта *дата* в строку даты и времени по Гринвичскому меридиану (GMT). Этот метод является устаревшим и поддерживается только в целях совместимости. Вместо него следует пользоваться методом **toUTCString**. Формат результирующей строки зависит от обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toGMTString());
```

выведет на экран обозревателя Internet Explorer строку вида `Tue, 26 Sep 2000 05:16:41 UTC`, а на экран Netscape Navigator — строку вида `Tue, 26 Sep 2000 05:16:41 GMT`.

Метод toLocaleDateString

Синтаксис: *дата*.toLocaleDateString()
Результат: строковое выражение
Поддержка: IE - Поддерживается с версии 5.5.
NN - Не поддерживается.

Метод **toLocaleDateString** преобразует примитивное значение объекта *дата* в строку даты по местному времени с учетом локальных установок операционной системы. Формат результирующей строки зависит от локализации операционной системы и обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toLocaleDateString());
```

в русифицированной версии Windows выведет на экран обозревателя Internet Explorer строку вида `26 сентября 2000 г.`

Метод toLocaleString

Синтаксис: *дата*.toLocaleString()
Результат: строковое выражение

Метод **toLocaleString** преобразует примитивное значение объекта *дата* в строку даты и времени по местному времени с учетом локальных установок операционной системы. Формат результирующей строки зависит от локализации операционной системы и обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toLocaleString());
```

в русифицированной версии Windows может вывести на экран обозревателя Internet Explorer строку вида `26 сентября 2000 г. 11:24:24`, а на экран Netscape Navigator — строку вида `Tuesday, September 26, 2000 11:24:24`.

Метод toLocaleTimeString

Синтаксис: *дата*.toLocaleTimeString()
Результат: строковое выражение
Поддержка: Поддерживается с версии 5.5.
Не поддерживается.

Метод **toLocaleTimeString** преобразует примитивное значение объекта *дата* в строку времени по местному времени с учетом локальных установок операционной системы. Формат результирующей строки зависит от локализации операционной системы и обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toLocaleTimeString());
```

в русифицированной версии Windows выведет на экран обозревателя Internet Explorer строку вида `11:24:24`.

Метод toTimeString

Синтаксис: *дата*.toTimeString()
Результат: строковое выражение
Поддержка: Поддерживается с версии 5.5.
Не поддерживается.

Метод **toTimeString** преобразует примитивное значение объекта *дата* в строку времени по местному времени. Например, сценарий

```
var d = new Date();
document.write(d.toTimeString());
```

выведет на экран обозревателя Internet Explorer строку вида `13:05:09 UTC+0500`.

Метод toString

Синтаксис: *дата*.toString()
Результат: строковое выражение

Метод **toString** преобразует примитивное значение объекта *дата* в строку даты и времени по местному времени. Формат результирующей строки зависит от обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toString());
```

может вывести на экран обозревателя Internet Explorer строку вида `Tue Sep 26 11:28:35 UTC-0700 2000` , а на экран Netscape Navigator — строку вида `Tue Sep 26 11:28:35 GMT-0700 (Pacific Daylight Time) 2000`.

Метод toUTCString

Синтаксис: *дата*.toUTCString()
Результат: строковое выражение

Метод **toUTCString** преобразует примитивное значение объекта *дата* в строку даты и времени по времени UTC. Формат результирующей строки зависит от обозревателя, например, сценарий

```
var d = new Date();
document.write(d.toUTCString());
```

выведет на экран обозревателя Internet Explorer строку вида `Tue, 26 Sep 2000 05:16:41 UTC`, а на экран Netscape Navigator — строку вида `Tue, 26 Sep 2000 05:16:41 GMT`.

Метод UTC

Синтаксис: Date.UTC(*год*, *месяц*, *день* [,*часы* [,*минуты* [,*секунды* [,*мс*]?]?]?]?)
Аргументы: *год*, *месяц*, *день*, *часы*, *минуты*, *секунды*, *мс* — числовые выражения
Результат: числовое значение

Метод **UTC** преобразует заданные дату и время в примитивное значение объекта **Date** по времени UTC и возвращает его. Аргументы имеют следующие значения:

- *год* — выражение, задающее полный номер года (например, 1988, а не 88);
- *месяц* — выражение, задающее номер месяца (0 = январь, 1 = февраль, …, 11 = декабрь);
- *день* — выражение, задающее номер для в месяце от 1 до 31;
- *часы* — необязательное выражение, задающее номер часа от 0 до 23;
- *минуты* — необязательное выражение, задающее номер минуты от 0 до 59;
- *секунды* — необязательное выражение, задающее номер секунды от 0 до 59;
- *мс* — необязательное выражение, задающее номер миллисекунды от 0 до 999.

Это статический метод объекта **Date**, поэтому для доступа к нему используется форма `Date.UTC`, а не *объект*`.UTC`. Он полностью аналогичен конструктору **Date** с единственным отличием: полученное примитивное значение вычисляется по времени UTC, а не по местному времени.

Этот метод полезен для присваивания значения объектам **Date**. Например, следующий сценарий

```
var x = Date.UTC(2000, 1, 2, 12, 35, 40);
var d = new Date(x);
document.write(d.toUTCString());
```

выведет на экран обозревателя строку `Wed, 2 Feb 2000 12:35:40 UTC`.

Метод valueOf

Синтаксис: *дата*.valueOf()
Результат: числовое значение

Метод **valueOf** возвращает примитивное значение объекта *дата*. Иными словами, этот метод возвращает тот же результат, что и метод **getTime**.