

Ch30 Date

Date는 날짜와 시간을 위한 메서드를 제공하는 빌트인 객체이면서 생성자 함수이다.
UTC(협정 세계시)는 국제 표준시를 말하고 GMT(그리니치 평균시)로 불리기도 한다. KST(한국 표준시간)는 UTC에 9시간을 더한 시간이다.

30.1 Date 생성자 함수

1. `new Date()`
 - 현재 날짜와 시간을 가지는 Date 객체를 반환한다.
 - Date객체는 내부적으로 날짜와 시간을 나타내는 정수값을 갖지만 출력시 기본적으로 날짜와 시간 정보를 출력한다.
 - new 연산자 없이 호출하면 객체가 아닌 문자열로 반환한다.

```
new Date(); // -> Mon Jul 06 2020 01:03:18 GMT+0900 (대한민국 표준시)
Date(); // -> "Mon Jul 06 2020 01:10:47 GMT+0900 (대한민국 표준시)"
```
2. `new Date(milliseconds)`
 - 밀리초를 인수로 전달하면 1970년 1월 1일 00:00:00(UTC)을 기점으로 전달된 밀리초만큼 경과된 날짜와 시간을 나타내는 Date객체를 반환한다.

```
/*
86400000ms는 1day를 의미한다.
1s = 1,000ms
1m = 60s * 1,000ms = 60,000ms
1h = 60m * 60,000ms = 3,600,000ms
1d = 24h * 3,600,000ms = 86,400,000ms
*/
```
3. `new Date(dateString)`
 - 날짜와 시간을 나타내는 문자열을 인수로 전달하면 지정된 날짜의 Date 객체를 반환한다.
 - 이때 전달한 문자열은 `Date.parse` 메서드에 의해 해석가능한 형식이어야한다.

```
new Date('May 26, 2020 10:00:00');
// -> Tue May 26 2020 10:00:00 GMT+0900 (대한민국 표준시)
```
4. `new Date(year, month[, day, hour, minute, second, millisecond])`
 - 연, 월, 일, 시, 분, 초, 밀리초를 인수로 전달하면 지정된 날짜와 시간을 나타내는 객체를 반환한다.
 - 연, 월은 반드시 지정해야 하며 지정하지 않은 옵션은 0 또는 1로 초기화된다.

인수	내용
year	연을 나타내는 1900년 이후의 정수. 0부터 99는 1900부터 1999로 처리됨
month	월을 나타내는 0 ~ 11까지의 정수(주의: 0부터 시작, 0 = 1월)
day	일을 나타내는 1 ~ 31까지의 정수
hour	시를 나타내는 0 ~ 23까지의 정수
minute	분을 나타내는 0 ~ 59까지의 정수
second	초를 나타내는 0 ~ 59까지의 정수
millisecond	밀리초를 나타내는 0 ~ 999까지의 정수

```
// 다음처럼 표현하면 가독성이 훨씬 좋다.  
new Date('2020/3/26/10:00:00:00');  
// -> Thu Mar 26 2020 10:00:00 GMT+0900 (대한민국 표준시)
```

30.2 Date 메서드

1. `Date.now`
 - 기준시간(1970년 1월 1일 00:00:00(UTC))을 기점으로 현재 시간까지 경과한 밀리초를 숫자로 반환한다.
2. `Date.parse`
 - 기준시간을 기점으로 인수로 전달된 지정 시간까지의 밀리초를 숫자로 반환한다.
3. `Date.UTC`
 - 기준시간을 기점으로 인수로 전달된 지정 시간까지의 밀리초를 숫자로 반환한다.
 - `new Date(year, month, ...)` 와 같은 형식의 인수를 사용해야한다.
 - 인수는 로컬 타임(KST)이 아닌 UTC로 인식된다.
4. `Date.prototype.getFullYear`
 - Date 객체의 연도를 나타내는 정수를 반환한다.
5. `Date.prototype.setFullYear`
 - Date 객체에 연도를 나타내는 정수를 설정한다. (연도 이외에 옵션으로 월,일도 설정할 수 있다)
6. `Date.prototype.getMonth`
 - Date 객체의 월을 나타내는 0 ~ 11의 정수를 반환한다.
7. `Date.prototype.setMonth`
 - Date 객체에 월을 설정한다. (옵션으로 일도 설정할 수 있다)
8. `Date.prototype.getDate`
 - Date 객체의 날짜를 나타내는 정수를 반환한다.
9. `Date.prototype.setDate`
 - Date 객체에 날짜를 설정한다.
10. `Date.prototype.getDay`
 - Date 객체의 요일(0 ~ 6)을 나타내는 정수를 반환한다. (일요일 0부터 토요일 6까지)
11. `Date.prototype.getHours`
 - Date 객체의 시간(0 ~ 23)을 나타내는 정수를 반환한다.
12. `Date.prototype.setHours`

- Date 객체에 시간을 설정한다. (옵션으로 분, 초, 밀리초도 설정할 수 있다)

13. `Date.prototype.getMinutes`

- Date 객체의 분(0 ~ 59)을 나타내는 정수를 반환한다.

14. `Date.prototype.setMinutes`

- Date 객체에 분을 설정한다. (옵션으로 초, 밀리초도 설정할 수 있다)

15. `Date.prototype.getSeconds`

- Date 객체의 초(0 ~ 59)를 나타내는 정수를 반환한다.

16. `Date.prototype.setSeconds`

- Date 객체에 초를 설정한다. (옵션으로 밀리초도 설정할 수 있다)

17. `Date.prototype.getMilliseconds`

- Date 객체의 밀리초(0 ~ 999)를 나타내는 정수를 반환한다.

18. `Date.prototype.setMilliseconds`

- Date 객체에 밀리초를 설정한다.

19. `Date.prototype.getTime`

- 기준시간을 기점으로 Date 객체의 시간까지 경과된 밀리초를 반환한다.

20. `Date.prototype.setTime`

- Date 객체에 기준시간을 기점으로 경과된 밀리초를 설정한다.

21. `Date.prototype.getTimezoneOffset`

- UTC와 Date 객체에 지정된 로컬 시간과의 차이를 분 단위로 반환한다.

22. `Date.prototype.toString`

- 사람이 읽을 수 있는 형식의 문자열로 Date 객체의 날짜를 반환한다.

23. `Date.prototype.toTimeString`

- 사람이 읽을 수 있는 형식의 문자열로 Date 객체의 시간을 반환한다.

24. `Date.prototype.toISOString`

- ISO 8601 형식으로 Date 객체의 날짜와 시간을 표현한 문자열을 반환한다.

```
const today = new Date('2020/7/24/12:30');

today.toString(); // -> Fri Jul 24 2020 12:30:00 GMT+0900 (대한민국 표준시)
today.toISOString(); // -> 2020-07-24T03:30:00.000Z
```

25. `Date.prototype.toLocaleString`

- 인수로 전달한 로캘을 기준으로 Date 객체의 날짜와 시간을 표현한 문자열을 반환한다.
- 인수가 생략되면 동작 중인 시스템의 로캘을 적용한다.

```
const today = new Date('2020/7/24/12:30');

today.toString(); // -> Fri Jul 24 2020 12:30:00 GMT+0900 (대한민국 표준시)
today.toLocaleString(); // -> 2020. 7. 24. 오후 12:30:00
today.toLocaleString('ko-KR'); // -> 2020. 7. 24. 오후 12:30:00
```

26. `Date.prototype.toLocaleTimeString`

- 인수로 전달한 로캘을 기준으로 Date 객체의 시간을 표현한 문자열을 반환한다.