

학습목표

- ✓자바스크립트 기본 용어를 알 수 있다.
- ✓기본 출력 방법을 습득하고 사용할 수 있다.
- ✓기본 자료형, 변수, 상수와 연산자를 이해하고 사용할 수 있다.
- ✓자료형을 변환할 수 있다.

CONTENTS

- + 기본 용어
- + 출력, 기본 자료형, 변수
- + 연산자
- + 자료형과 자료형 검사
- + 자료형 변환

1. 기본 용어

- 표현식과 문장

- 표현식 (expression)

```
365  
10 * 2 + 20  
"JavaScript"
```

- 문장 (statement)

- 하나 이상의 표현식이 모이면 문장이 됨
 - 문장의 마지막에는 ;(세미콜론)으로 마무리 함



1. 기본 용어

- 표현식과 문장

- 프로그램 (program)

- 문장이 모여 프로그램이 됨.

```
let a = 365;  
let b = 10 * 2 + 20;  
console.log( a + b );
```



1. 기본 용어

• 키워드

– 자바스크립트에서 특별한 의미가 부여된 단어

break	else	instanceof	true
case	false	new	try
catch	finally	null	typeof
continue	for	return	var
default	function	switch	void
delete	if	this	while
do	in	throw	with

1. 기본 용어

• 식별자

– 이름을 붙일 때 사용하는 단어, 변수와 함수 이름 등으로 사용

- 키워드(예약어) 사용 불가
- 특수 문자는 \$와 _만 허용
- 숫자로 시작 불가
- 공백은 입력 불가

omega, omega1, \$omega, _omega

1. 기본 용어

• 식별자

– 식별자 사용 규칙

- 생성자 함수의 이름은 항상 대문자로 시작
- 변수, 함수, 속성, 메소드의 이름은 항상 소문자로 시작
- 여러 단어로 된 식별자는 각 단어의 첫 글자를 대문자로 함

run away	➔	runAway
just do it	➔	justDoIt

1. 기본 용어

구분	앞에 . 없음 (단독 사용)	앞에 . 있음 (함께 사용)
뒤에 () 없음	변수	속성
뒤에 () 있음	함수	메소드

사용 예	종류
trans(200)	함수
Array.length	속성
Input	변수
msg('Message', 'Defstr')	함수
Math.PI	속성
console.log('console print')	메소드

1. 기본 용어

- 주석
 - 프로그램의 진행에 영향을 주지 않는 코드

방법	표현
한 줄 주석	// 주석 내용
여러 줄 주석	/* 주석 내용 주석 내용 */

2. 출력

- 출력 메소드
 - console 객체의 log () 메소드 사용 : console.log ()

```
console.log( '문자열' );
```

2. 출력

- REPL을 사용한 출력
 - 곧바로 문장을 입력해서 출력

```
d:\> node
> “안녕” + “자바스크립트”
‘안녕자바스크립트’
> 365 + 35
400
>
```



3. 변수 (Variable)

- 변수
 - 값을 저장할 때 사용하는 식별자
 - 모든 자료형의 저장이 가능
- 변수의 사용 방법
 - 변수 선언 : 변수를 만드는 것
 - 변수 할당 : 변수에 값을 저장(대입)하는 것
 - 변수 초기화 : 선언과 할당을 한 번에 하는 것



3. 변수 (Variable)

- 변수 선언

```
var radius;    //ECMAScript5  
let color;     //ECMAScript6
```

- 변수 할당

```
var color;  
color = 'green';
```

- 변수 초기화

```
var color = 'green';
```



4. 기본 자료형

- 문자열 (String)

- 문자의 집합
- 문자열 생성시 큰따옴표나 작은따옴표를 사용
- 문자열을 만드는 방법
 - 큰 따옴표 사용 : "동해물과 백두산이"
 - 작은 따옴표 사용 : '동해물과 백두산이'
 - 두 가지 중 어떤 방법으로 문자열을 만들어도 상관없으나, 일관되게 사용

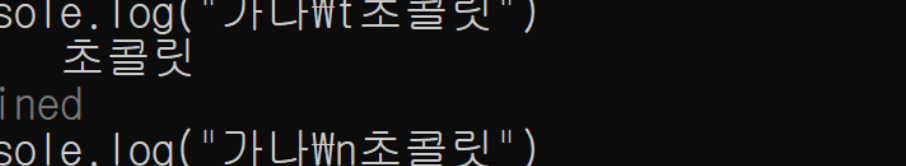


4. 기본 자료형

- 이스케이프 문자

- 따옴표를 문자 그대로 사용 가능
- 문자열을 줄 바꿈 할 경우 사용

이스케이프 문자	설명
Wt	수평 탭
Wn	줄 바꿈
W'	작은 따옴표
W''	큰 따옴표
WW	역 슬래시



The screenshot shows a Node.js console window with the following commands and output:

```

> console.log("가나Wt초콜릿")
가나      초콜릿
undefined
> console.log("가나Wn초콜릿")
가나
초콜릿
undefined
> console.log("W"가나W" 초콜릿")
"W가나" 초콜릿
undefined
>

```

4. 기본 자료형

– 문자열 연결 연산자

연산자	설명
+	문자열 연결

- '가나다' + '라마' + '바사아' + '자차카타' + '파하'
➔ 가나다라마바사아자차카타파하

4. 기본 자료형

– 문자 선택 연산자

연산자	설명
문자열[숫자]	문자 선택

```
가
undefined
> console.log("가나초콜릿"[1])
나
undefined
> console.log("가나초콜릿"[2])
초
undefined
> console.log("가나초콜릿"[3])
콜
undefined
> console.log("가나초콜릿"[4])
릿
undefined
>
```

4. 기본 자료형

- 숫자
 - 가장 기본적인 자료형
 - 정수와 실수의 구분 없이 숫자는 모두 숫자
 - 기본 사칙 연산자

연산자	설명
+	덧셈
-	뺄셈
*	곱셈
/	나눗셈

4. 기본 자료형

- 숫자
 - 나머지 연산자

연산자	설명
%	나머지

```
관리자: C:\Windows\system32\cmd.exe - node
> 7 % 3
1
> 
```

4. 기본 자료형

• 비교 연산자

– 두 대상을 비교하여 참/거짓을 판단하는 연산자

연산자	설명
>=	좌변이 우변보다 크거나 같다
<=	좌변이 우변보다 작거나 같다
>	좌변이 크다
<	우변이 크다
==	좌변과 우변이 같다
!=	좌변과 우변이 같지 않다(다르다)

4. 기본 자료형

• 불리언(Boolean)

– 참/거짓에 대한 **논리값**(True 또는 False)을 표현할 때 사용

- 50 > 200 → false
- 50 < 200 → true

```
C:\Windows\System32\cmd.exe
> 3 > 5
false
> 3 < 5
true
>
```

• 논리 연산자

연산자	설명
!	논리 부정
&&	논리 곱
	논리 합

```
C:\Windows\System32\cmd.exe - node
> true
true
> !true
false
> false
false
> !false
true
>
```

4. 기본 자료형

- 논리곱 연산자 (&&)

- 좌변과 우변이 모두 참일 때만 참

좌변	우변	결과
true	true	true
true	false	false
false	true	false
false	false	false

```

관리자: C:\Windows\system32\cmd.exe - node
true
> true && true
true
> true && false
false
> false && false
false
> false && true
false
> 3 > 0 && 2 > 1
true
>
    
```

4. 기본 자료형

- 논리합 연산자 (||)

- 좌변과 우변이 모두 거짓일 때만 거짓

좌변	우변	결과
true	true	true
true	false	true
false	true	true
false	false	false

```

관리자: C:\Windows\system32\cmd.exe - node
true
> true || true
true
> true || false
true
> false || true
true
> false || false
false
> 2 < 1 || 1 < 0
false
>
    
```

5. 복합 대입 연산자

- 대입 연산자와 다른 연산자를 함께 사용하는 연산자

연산자	설명		
+=	기존 변수의 값에 값을 더한 후 대입	a += b	a = a + b
-=	기존 변수의 값에 값을 뺀 후 대입	a -= b	a = a - b
*=	기존 변수의 값에 값을 곱한 후 대입	a *= b	a = a * b
/=	기존 변수의 값에 값을 나눈 후 대입	a /= b	a = a / b
%=	기존 변수의 값의 나머지를 구하여 대입	a %= b	a = a % b

6. 증감 연산자

- 복합 대입 연산자를 간략하게 사용한 형태
- 다른 연산자나 함수와 함께 사용할 때 차이가 있음

연산자	설명
변수++	기존 변수 사용 후 변수의 값 1 증가
++변수	기존 변수의 값을 1 증가 시킨 후 변수 사용
변수--	기존 변수 사용 후 변수의 값 1 감소
--변수	기존 변수의 값을 1 감소 시킨 후 변수 사용

7. 자료형 검사

- 자료형을 확인하는 연산자

연산자	설명
typeof	변수의 자료형을 추출하여 나타냄

```
관리자: C:\Windows\system32\cmd.exe - node
> typeof 5
'number'
> typeof 'javascript'
'string'
> typeof true
'boolean'
>
```

8. undefined

- 선언은 되었으나 할당이 되지 않은 자료형

```
> let color
```

```
관리자: C:\Windows\system32\cmd.exe - node
> let color
undefined
> var width
undefined
>
```

9. 강제 자료형 변환

• 강제 자료형 변환 함수

함수	설명
Number()	숫자 자료형으로 변환
String()	문자열 자료형으로 변환
Boolean()	논리 자료형으로 변환

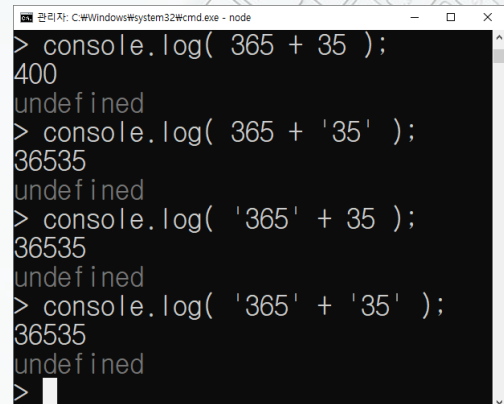
- '숫자로 변환할 수 없는 문자열'을 Number() 함수로 변환하면 'NaN' 출력
 - NaN(Not a Number)은 '숫자 자료형이지만 숫자가 아닌 것'을 의미
- Boolean() 함수를 사용하면 5개의 요소는 false로 변환
 - 0, NaN, "", null, undefined

10. 자동 자료형 변환

• 숫자와 문자열 자료형 자동 변환

- 숫자와 문자열에 '+' 연산자를 적용하면 숫자는 자동으로 문자열로 변환됨

```
console.log( 365 + 35 );
console.log( 365 + '35' );
console.log( '365' + 35 );
console.log( '365' + '35' );
```



```
> console.log( 365 + 35 );
400
undefined
> console.log( 365 + '35' );
36535
undefined
> console.log( '365' + 35 );
36535
undefined
> console.log( '365' + '35' );
36535
undefined
>
```


10. 자동 자료형 변환

- 논리 자료형 자동 변환

– ! 연산자를 두 번 쓰면 Boolean() 함수를 사용하는 것과 같음

```
console.log( !!0 );
console.log( !!null );
```

```
관리자: C:\Windows\system32\cmd.exe - node
> !!0
false
> !!null
false
> !!undefined
false
>
```

11. 일치 연산자

- 일치 연산자

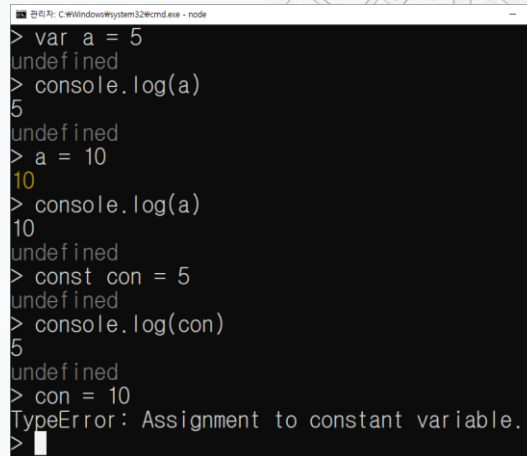
– 값은 물론 자료형까지 검사하여 참/거짓을 판단함

연산자	설명
===	양 변의 자료형과 일치하는 값
!==	양 변의 자료형과 일치하지 않는 값

```
관리자: C:\Windows\system32\cmd.exe - node
> 5 == 5
true
> 5 == '5'
true
> 5 === '5'
false
>
```

12. 상수 (Constant)

- '항상 같은 수'라는 의미, 변수와 반대되는 개념
- `const` : 상수constant를 만드는 키워드
- 변하지 않을 대상에 상수를 적용



```
C:\Windows\System32\cmd.exe - node
> var a = 5
undefined
> console.log(a)
5
undefined
> a = 10
10
> console.log(a)
10
undefined
> const con = 5
undefined
> console.log(con)
5
undefined
> con = 10
TypeError: Assignment to constant variable.
>
```