

JavascriptⅡ: 흐름 제어

Practical Web Programming

김현준

Overview of today's lecture

- Lecture #1 review
- Logical operations
- *if* statement
- *while* statement
- *for* statement
- *continue, break* statements
- Variable scope
- Small project: *Prime Finder*
- Preview for the next lecture

Review: 정보와 연산자

정보에는 숫자와 문자가 있다.

- 숫자 표기 예시: *36.5* *-273* *3.141592* *1/2*
true *false*
- 문자 표기 예시: *"New York"* *"6724"* *'I'm Paul'*

연산자를 통해 정보를 조작할 수 있다.

- 수 연산자: *+* *-* */* *** *%* *&&* *//* *!* *등등*
- 문자 연산자: *+*

같은 형식의 데이터끼리 연산자를 사용할 수 있다.

- 필요할 경우 *캐스팅(casting)* 을 통해 상호 변환할 수 있다.

Review: 정보와 연산자

1. 사칙 연산 우선순위가 적용됨
2. 괄호는 () 한 종류만 사용할 수 있다.
3. 수 - 문자 캐스팅 : *String()*
4. 문자 - 수 캐스팅 : *Number()*
경우 예러

* 비 숫자 문자가 들어갈

캐스팅 예시: *"M" + String(6723 + 1) + " BUS"* 는 *"M6724 BUS"* 와 동일

Review: 변수

변수

- 데이터는 가변적일 때가 많다.
- 데이터를 저장하였다가 필요 시 읽고 쓸 수 있게 해 준다.
- 데이터를 담는 '컵'
- **var** 명령을 통해 선언할 수 있으나 생략 가능하다.

변수에 데이터 담기

변수명 = 데이터
위치해야 한다.

$Pi = 3.141592$

$Tau = Pi * 2$

* 변수명이 반드시 왼쪽 항에

Review: 변수

- 변수명을 지을 땐,
 1. 중간에 띄어쓰기 안 됨
 2. 특수문자 안 됨, 그러나 한글과 언더바(_)는 됨
 3. 숫자로 변수명을 시작할 수 없음
- 변수 안에 담긴 데이터의 종류에 따라 해당하는 모든 연산자 사용 가능
- 변수에 값을 대입할 때는 = 연산자 사용

Review: 함수

- 특정 동작을 수행하는 일정 코드 부분을 묶어 놓은 것
- 수학에서의 함수 개념과 유사, 입력을 받아 출력을 내놓는다.

함수 형식 (함수의 값은 함수의 반환값)

$f(x, y, z, \dots)$

예시

데이터 출력: *alert(데이터)*

데이터 입력: *prompt (제목)*

Review: 함수

```
function 함수명 ( 매개 변수 ) {  
    return 함수값  
}
```


논리 연산

두 값의 대소 관계, 일치 여부 등을 판단하는 연산자

- > >= < <=
- == !=

논리값 사이의 관계를 판단하는 연산자

- &&
- ||
- !

논리 연산의 결과값은 1(true) 나 0(false)로 나옴

논리 연산

$1 + 1 > 2$ = 0

$1 + 1 >= 2$ = 1

$"M6724" != "M" + String(6724)$ = 0

$A = 3$ 일 때,

$A > 2 \&\& A < 3$ = 0

$A > 2 \// A < 3$ = 1

$A > 2 \&\& A <= 3$ = 1

논리 연산자는 우선순위가 낮다. (사칙연산 $>$ ($>=$ $>$ $<=$ $<$) $>$ ($\&\&$ $\//$))

논리 연산

키와 몸무게를 입력받아 체지방 지수를 계산하고 그 사람이 비만인지 여부(1혹은 0)를 출력하는 프로그램을 작성해 봅시다.

- BMI = 몸무게 / (키 * 키)
- BMI가 30 이상이면 비만

if 문

- 특정 조건이 만족될 때에만 명령이 실행되도록 한다.

```
if ( 조건 ) {  
    조건이 참(=1) 일 경우 실행되는 명령  
}
```

if 문

```
Height = prompt("당신의 키는?")  
Mass = prompt("당신의 몸무게는?")  
BMI = Height / (Mass * Mass)
```

```
If ( BMI >= 30 ) {  
    alert( "비만입니다" )  
}
```

if-else 문

- 비만이 아닐 경우에 무언가가 실행되게 하려면?
- else 문은 if 문과 쌍으로 사용될 수 있다.

```
if ( 조건 ) {  
    조건이 참일 경우  
} else {  
    조건이 거짓일 경우  
}  
...
```

if-else 문

```
If ( BMI >= 30 ) {  
    alert( "비만입니다" )  
} else {  
    alert( "비만이 아닙니다" )  
}
```

if-if else문

- 저체중, 과체중 여부도 알고 싶다!

```
if ( BMI >= 30 ) {  
    alert( "비만입니다" )  
} else {  
    if ( BMI >= 25) {  
        alert("경도비만입니다.");  
    } else {  
        if ( BMI >= 20) {  
            alert("정상입니다.");  
        } else {  
            alert("저체중입니다.");  
        }  
    }  
}
```

길다!

if-if else문

```
If ( BMI >= 30 ) {  
    alert( "비만입니다" )  
} else if (BMI >= 25){  
    alert( "경도비만입니다." )  
} else if (BMI >= 20){  
    alert( "정상입니다." )  
} else {  
    alert( "저체중입니다." )  
}
```

if - if else - else 문

- 여러분의 중간고사 점수를 넣으면 학점을 반환해 주는 프로그램을 만들어 보세요.
- 입력: 0~ 100 사이의 점수
- 학점 기준: A: 90이상, B: 80 이상, C: 70 이상, D: 60이상, F: 60 미만

while 문

- 조건이 참이면 계속 반복하여 실행됨
- 조건이 변하지 않으면 한번도 실행되지 않거나, 무한반복(으아아) 되므로 조건을 변화시켜 주어야 함

while (조건) {

조건이 참일 경우 실행

}

while 문

나이 = 20

while (나이 < 100) {

alert("100세 미만은 금연입니다.")

나이 = 나이 + 1

}

while 문

반복할횟수 = prompt("몇번 반복할까요?")

지금반복횟수 = 0

while (지금반복횟수 < 반복할횟수) {

지금반복횟수 ++

alert (지금반복횟수 + " 번째 반복중입니다.")

}

while 문

- 구구단을 출력하는 프로그램을 만들어 보세요
- 2단부터 9단까지 출력합니다.

for 문

- While 문을 보면, 형식이 정형화 되어 있다.
- 처음 조건을 초기화 해 주는 부분 = A
- 조건을 업데이트 해 주는 부분 = B

```
for ( A;   조건;   B ) {  
    조건이 참일 경우 실행되는 명령  
}
```

for 문

나이 = 20

while (나이 < 100) {

alert("100세 미만은 금연입니다.")

나이 = 나이 + 1

}

for (나이 = 20; 나이 < 100; 나이++) {

alert("100세 미만은 금연입니다.")

}

for 문

for문을 사용하여 다시 한 번 구구단을 짜 보세요.

continue, break 문

이번 턴은 그냥 패스하고 다음 턴 시작할래! : continue
반복문 여기서 그만 종료할래! : break

```
for ( i= 0; i < 10; i ++ ) {  
    if ( i == 3 ) {  
        continue / break  
    }  
    alert( i )  
}
```

Small Project: 소수 찾기

- 소수? 1 과 자기 자신만으로 나누어 떨어지는 자연수
- 2, 3, 5, 7, 11, 13, 17 ...

1. 임의의 자연수를 입력받고, 이 수가 소수인지 판별해 주는 프로그램을 작성해 보세요.
2. 1부터 100,000,000 사이에는 소수가 몇개 있을까요?