

# ABC3012: Advanced Web Programming

Donghyun Kang ([donghyun@changwon.ac.kr](mailto:donghyun@changwon.ac.kr))

NOSLab (<https://noslab.github.io>)

Changwon National University

# switch 문

- switch 문

- 값에 따라 서로 다른 코드를 실행할 때, switch 문 적합

```
switch(식) {  
  case 값1: // 식의 결과가 값1과 같을 때  
    실행 문장 1;  
    break;  
  case 값2: // 식의 결과가 값2와 같을 때  
    실행 문장 2;  
    break;  
  ...  
  case 값m:  
    실행 문장 m; // 식의 결과가 값과 같을 때  
    break;  
  default: // 어느 값과도 같지 않을 때  
    실행 문장 n;  
}
```

```
var fruits="사과";  
switch(fruits) {  
  case "바나나":  
    price = 200; break;  
  case "사과":  
    price = 300; break;  
  case "체리":  
    price = 400; break;  
  default:  
    document.write("팔지 않습니다.");  
    price = 0;  
}  
  
// switch 문의 실행 결과 price=300
```

# case 문의 '값'

- case 문의 '값'은 상수(리터럴)만 가능
  - 잘 작성된 case 문

```
case 1 :  
case 2.7 :  
case "Seoul" :  
case true :  
case 2+3 : // 2+3은 먼저 5로 계산되어 case 5:와 동일
```

- case 문의 '값'에 변수나 식은 사용 불가
  - 잘못 작성된 case 문

```
case a : // 오류. 변수 a 사용 불가  
case a > 3 : // 오류. 식(a > 3) 사용 불가
```

# switch 문에서 break 문의 역할

- break 문
  - switch 문 종료
    - break; 문을 만날 때까지 아래로 코드 계속 실행

```
var city="Seoul";
switch(city) {
  case "Seoul":
    document.write("서울");
    break;
  case "NewYork":
    document.write("뉴욕");
    break;
  case "Paris":
    document.write("파리");
    break;
}
```

서울뉴욕

(a) break;를 만날 때까지 아래로 실행을  
계속하는 사례

```
var day="월";
switch(day) {
  case "월":
  case "화":
  case "수":
  case "목":
  case "금": document.write("정상영업");
             break;
  case "토":
  case "일": document.write("휴일");
             break;
}
```

정상영업

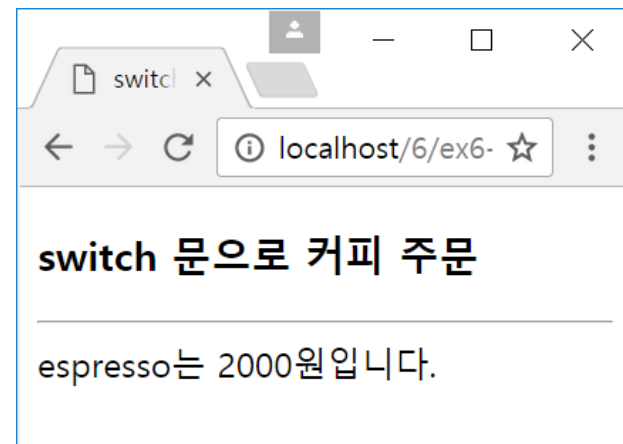
(b) 여러 case에 대해 동일한 코드를 실행하도록  
의도적으로 break;를 생략한 경우

# 예제 6-16 switch 문 사용

```
<!DOCTYPE html>
<html>
<head><title>switch</title></head>
<body>
<h3>switch 문으로 커피 주문</h3>
<hr>
<script>
  var price = 0;
  var coffee = prompt("무슨 커피 드릴까요?", "");
  switch(coffee) {
    case "espresso" :
      case "에스프레소" : price = 2000;
        break;
    case "카푸치노" : price = 3000;
      break;
    case "카페라떼" : price = 3500;
      break;
    default :
      document.write(coffee + "는 없습니다.");
  }
  if(price != 0)
    document.write(coffee + "는 " + price + "원입니다.");
</script>
</body>
</html>
```

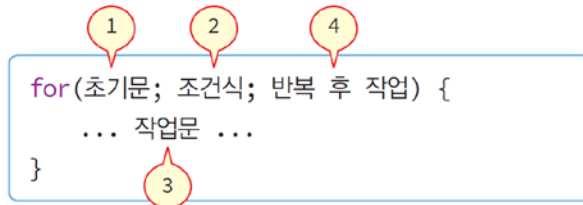
localhost 내용: ✕

무슨 커피 드릴까요?



# 반복문

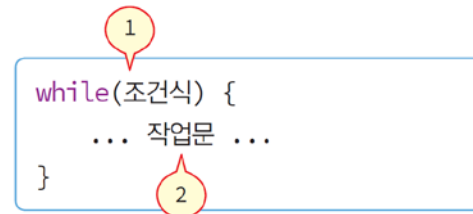
- for 문



```
// 0에서 9까지 출력  
for(var i=0; i<10; i++) {  
    document.write(i);  
}
```

0123456789

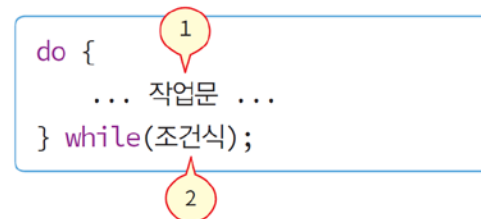
- while 문



```
var i=0;  
while(i<10) { // i가 0에서 9까지 출력  
    document.write(i);  
    i++;  
}
```

0123456789

- do-while 문

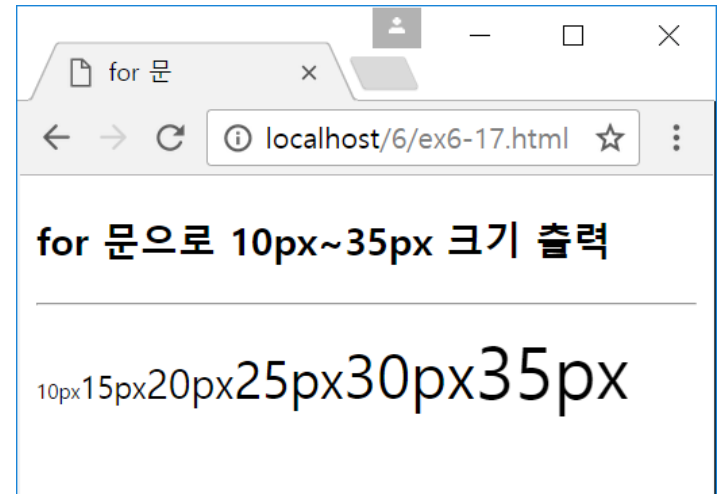


```
var i=0;  
do { // i가 0에서 9까지 출력  
    document.write(i);  
    i++;  
} while(i<10);
```

0123456789

# 예제 6-17 for 문으로 10px~35px 크기로 출력

```
<!DOCTYPE html>
<html>
<head>
<title>for 문</title>
</head>
<body>
<h3>for 문으로 10px~35px 크기 출력</h3>
<hr>
<script>
  for(var size=10; size<=35; size+=5) { // 5씩 증가
    document.write("<span ">
    document.write("style='font-size:" + size + "px'>");
    document.write(size + "px");
    document.write("</span>");
  }
</script>
</body>
</html>
```



# 예제 6-18 while 문으로 0~n까지의 합 구하기

```
<!DOCTYPE html>
<html>
<head>
<title>while 문</title>
</head>
<body>
<h3>while 문으로 0에서 n까지 합</h3>
<hr>
<script>
  var n = prompt("0보다 큰 정수를 입력하세요", 0);
  n = parseInt(n); // 문자열 n을 숫자로 바꿈

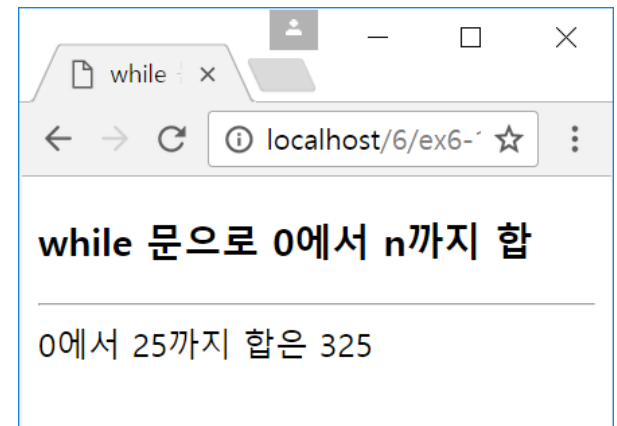
  var i=0, sum=0;
  while(i<=n) { // i가 0에서 n까지 반복
    sum += i;
    i++;
  }
  document.write("0에서 " + n + "까지 합은 " + sum);
</script>
</body>
</html>
```

prompt()가 리턴한 것은 문자열

localhost 내용: ✕

0보다 큰 정수를 입력하세요

확인 취소





# 예제 6-19 do-while 문으로 0~n까지 합 구하기

```
<!DOCTYPE html>
<html>
<head>
<title>do-while 문</title>
</head>
<body>
<h3>do-while 문으로 0에서 n까지 합</h3>
<hr>
<script>
  var n = prompt("0보다 큰 정수를 입력하세요", 0);
  n = parseInt(n); // 문자열 n을 숫자로 바꿈

  var i=0, sum=0;
  do {
    sum += i;
    i++;
  } while(i<=n); // i가 0~n까지 반복
  document.write("0에서 " + n + "까지 합은 " + sum);
</script>
</body>
</html>
```

prompt()가 리턴한 것은 문자열

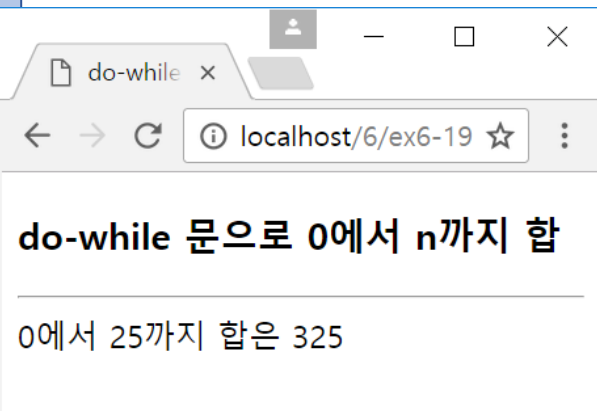
localhost 내용:

0보다 큰 정수를 입력하세요

25

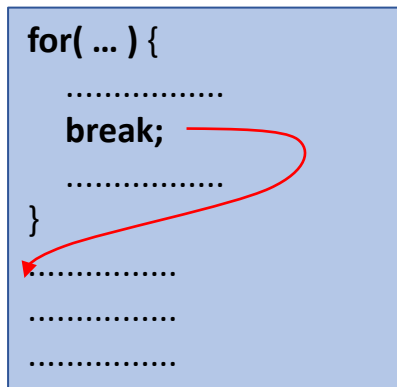
확인

취소

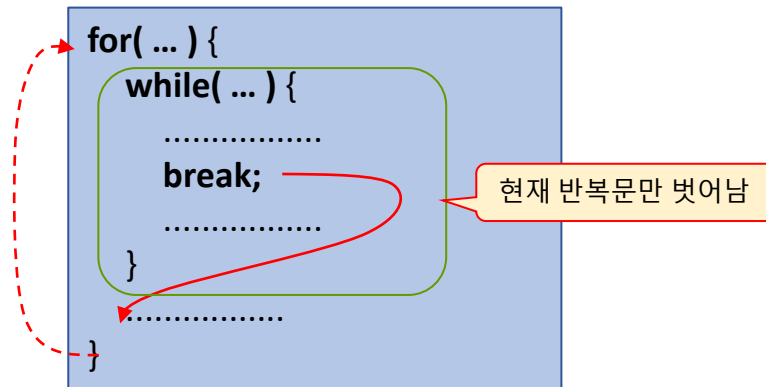


# 반복문 내의 break 문과 continue 문

- break 문 : 가장 안쪽 반복문 하나만 벗어나도록 제어

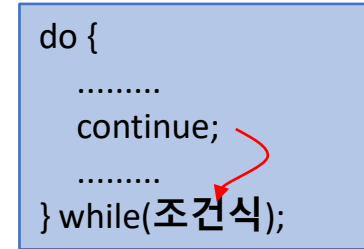
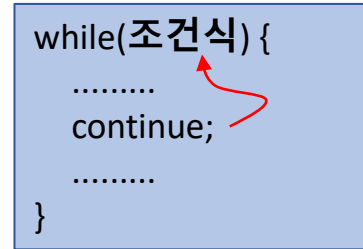
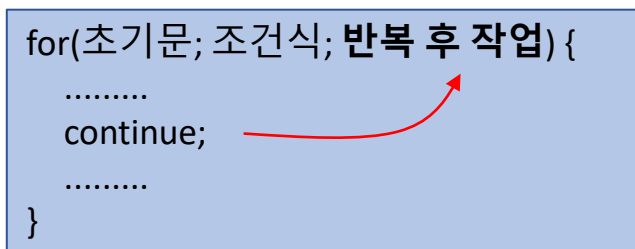


(a) 반복문 벗어나기



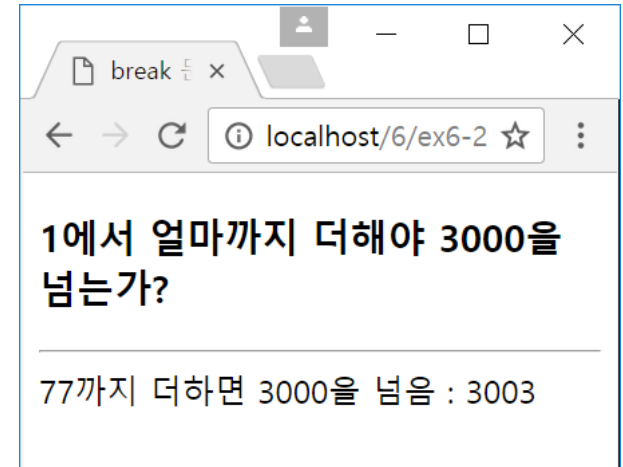
(b) 중첩 반복에서 현재 반복문만 벗어나

- continue 문 : 반복 코드 실행 중단, 다음 반복으로 점프



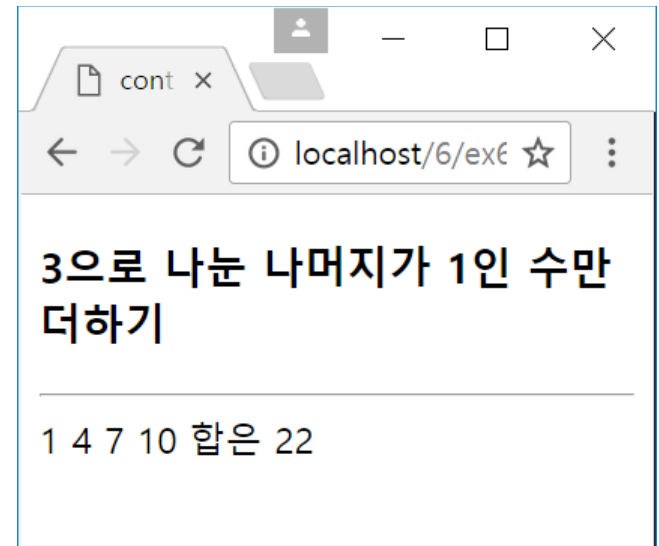
# 예제 6-20 break 문

```
<!DOCTYPE html>
<html>
<head>
<title>break 문</title>
</head>
<body>
<h3>1에서 얼마까지 더해야 3000을 넘는가?</h3>
<hr>
<script>
  var i=0, sum=0;
  while(true) { // 무한 반복
    sum += i;
    if(sum > 3000)
      break; // 합이 3000보다 큼. 반복문 벗어남
    i++;
  }
  document.write(i + "까지 더하면 3000을 넘음 : " + sum);
</script>
</body>
</html>
```



# 예제 6-21 continue 문

```
<!DOCTYPE html>
<html>
<head>
<title>continue 문</title>
</head>
<body>
<h3>3으로 나눈 나머지가 1인 수만 더하기</h3>
<hr>
<script>
  var sum=0;
  for(i=1; i<=10; i++) { // i가 1에서 10까지 반복
    if(i%3 != 1) // 3으로 나눈 나머지가 1이 아닌 경우
      continue; // 다음 반복으로 점프(i++ 코드로)
    document.write(i + " ");
    sum += i;
  }
  document.write("합은 " + sum);
</script>
</body>
</html>
```



# 함수

- 함수란?
  - 목적을 가지고 작성된 코드 블록
  - 데이터 전달받아 처리한 후 결과를 돌려주는 코드 블록
- 함수 개념



# 함수의 구성과 호출

- 함수의 구성

```
function 함수이름(arg1, arg2,..., argn) {  
    ...프로그램 코드...  
    결과를 리턴하는 return 문  
}
```

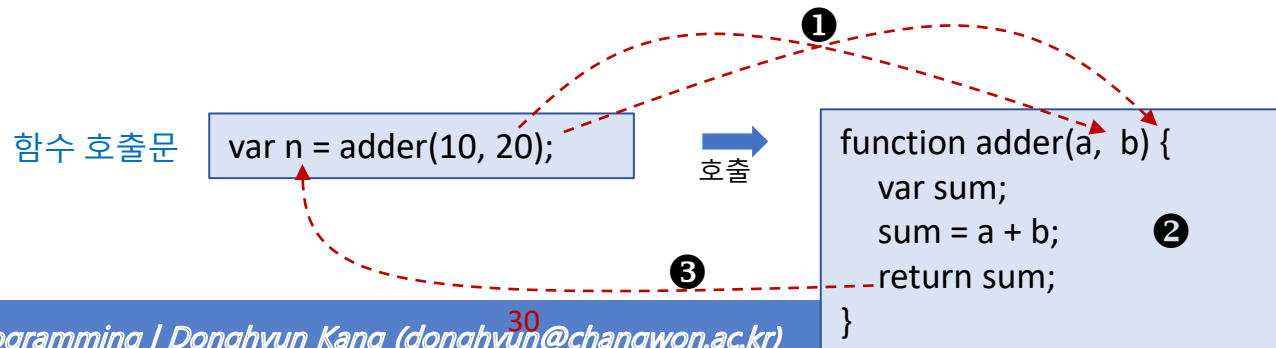
함수 선언      함수 이름   매개 변수

```
function adder (a, b) {  
    var sum;  
    sum = a + b;  
    return sum; // 덧셈 합 리턴  
}
```

반환 키워드      반환 값

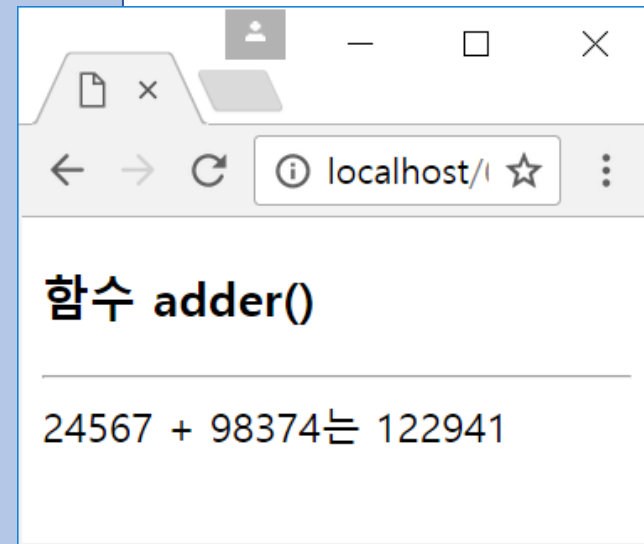
- 함수 호출

- 함수의 코드 실행 요청



# 예제 6-22 adder() 함수 작성 및 호출

```
<!DOCTYPE html>
<html>
<head>
<title>함수</title>
<script>
function adder(a, b) { // 함수 작성
    var sum;
    sum = a + b;
    return sum;
}
</script>
</head>
<body>
<h3>함수 adder()</h3>
<hr>
<script>
    var n = adder(24567, 98374); // 함수 호출
    document.write("24567 + 98374는 " + n + "<br>");
</script>
</body>
</html>
```



# Q&A