문장(Statement)

문장

- 형태: ; (세미콜론)
- 문장을 끝나게 하는 역할
  - var book = "책";

화이트 스페이스

- White Space
- 사람 눈에 보이지 않는 문자
  - 가독성을 위한 것
  - 문자마다 기능을 갖고 있음

White Space

Unicode	Unicode Name	약칭
₩u0009	수평 탭(Horizontal Tab)	<TAB>
₩u000B	수직 탭(Vertical Tab)	<VT>
₩u000C	폼 넘기기(Form Feed)	<FF>
₩u0020	공백(Space)	<SP>
₩u00A0	NBSP(No-break space)	<NBSP>
₩uFFF	BOM(Byte Order Mark)	<BOM>

Line Teminator

Unicode	Unicode Name	약칭
₩u000A	Line Feed(LF)	<LF>
₩u000D	Carriage Return(CR)	<CR>
₩u2028	Line Separator(LS)	<LS>
₩u2029	Paragraph Separator	<PS>

세미콜론 자동 삽입

- 세미콜론(;)은 문장 끝에 작성
- 세미콜론을 삽입하여
  - 자동으로 문장이 끝나게 합니다.
  - JS 엔진이 분석 및 삽입 코드 1

블록

- 형태: {문장 리스트opt} 코드 1
- 중괄호 {}
  - 실행 그룹으로
  - 블록 안의 모든 문장 실행
- 문장 리스트
  - {} 안의 모든 문장
  - 문장 리스트 작성은 선택(option)
  - 강좌에서 option를 opt로 표기

if, debugger

if

- 형태:
  if (표현식) 문장1
  if (표현식) 문장1 else 문장2
- 조건에 따른 처리
  - 먼저 표현식을 평가
  - 평가 결과를 true/false로 변환
  - true이면 문장1 실행 코드 1 코드 2
  - false이면 문장2 실행 코드 3 코드 4

debugger

- debugger 위치에서 실행 멈춤
  - 브라우저의 개발자 도구 창이 열려 있을 때만 멈춤
  - 열려있지 않으면 멈추지 않음
  - ES5부터 지원 코드 1
  - debugger 실행

while, do-while

while

- 형태: while (표현식) 문장
- 표현식의 평가 결과가 false가 될 때까지
  - 문장을 반복하여 실행
  - 반복이 종료되는 조건 필요 코드 1

do ~ while

- 형태: do 문장 while (표현식)
- 처리 방법은 while 문과 같음
  - 단, do 문을 먼저 실행 코드 1

for()

for()

- 형태: for (초깃값opt; 비교opt; 증감opt) 문장
- 2번째의 비교 표현식의 평가 결과가 true인 동안 문장을 반복 실행 코드 1

for() 옵션

- 형태: for (초깃값opt; 비교opt; 증감opt) 문장
- 형태에서 opt는 생략 가능
  - 증감 생략 코드 1
  - 초깃값과 증감 생략 코드 2
  - 비교와 증감 생략 코드 3
  - 모두 생략 코드 4

[코딩 시간]

- for() 문을 사용하여
  - 1에서 50까지 반복하면서
  - 홀수 번째 값과
  - 짝수 번째 값을 누적하고
  - 반복한 값을 누적합니다.
- 반복을 완료하면
  - 누적한 홀수 번째 값과
  - 누적한 짝수 번째 값을 출력합니다.
  - 누적한 전체 값을 출력합니다.

break, continue

break

- 형태:
  break;
  break 식별자;
- 반복문 종료 코드 1 코드 2
- for, for~in, while,
  do~while, switch에서 사용

continue

- 형태:
  continue;
  continue 식별자;
- 반복문의 처음으로 분기 코드 1
- for, for~in, while, do~while에서 사용

switch

switch

- 형태:
  switch (표현식) {
    case 표현식: 문장 리스트opt
    default: 문장 리스트opt
  };
- switch 표현식의 평가 값과 일치하는 case 문 수행 코드 1
- break 사용 코드 2
- 일치하는 case가 없으면 default 수행 코드 3
- OR(||) 형태 코드 4

try-catch, throw

try-catch

- 형태:
  try 블록 catch (식별자) 블록
  try 블록 finally 블록
  try 블록 catch (식별자) 블록 finally 블록
- try 문에서 예외 발생을 인식
- 예외가 발생하면 catch 블록 실행 코드 1
- finally 블록은
  예외 발생과 관계없이 실행 코드 2

throw

- 형태: throw 표현식;
- 명시적으로 예외를 발생시킴
- 예외가 발생하면 catch 실행

코드 1 코드 2 코드 3

strict 모드

strict 모드

- 형태: "use strict"
- 엄격하게 JS 문법 사용의 선언
- 작성한 위치부터 적용 코드 1 코드 2
- ES5부터 지원

[코딩 시간]

- JS 프로그램에서 사용하지 않는 문장이 있습니다.
  - label 문장의 코드를 작성하고 사용하지 않는 이유를 설명하세요.
  - 강좌에서 다루지 않았습니다.
- "use strict" 아래에 with 문을 사용한 코드를 작성하세요.
  - 강좌에서 다루지 않았습니다.
  - 에러가 발생하는 것을 확인하세요.
  - 에러가 발생하는 이유는 단계적 설명이 필요하므로 중고급 과정에서 다룹니다.