자바스크립트 기본

빅데이터 오케스트레이션 및 시각화 실습 – 1일차

학습 내용

1. 자바스크립트 개요
2. 입력과 출력
3. 변수와 연산자
4. 조건과 반복
5. 내장 객체
6. 함수
7. 스트링

실습 환경 구축

크롬 브라우저 설치

https://www.google.com/intl/ko_kr/chrome/

크롬 브라우저 개발자 도구

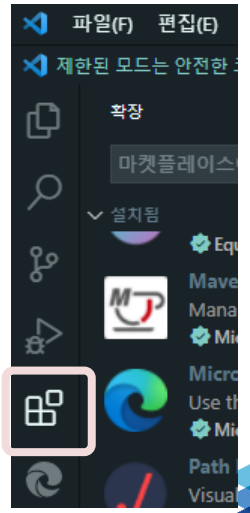
- ctrl + shift + i
- F12

Visual Studio Code 설치

https://code.visualstudio.com/

Visual Studio Code Extension

- Korean Language Pack for Visual Studio Code
- Live Server
- Auto Close Tag
- Auto Rename Tag
- Prettier
- Tailwind CSS IntelliSense
- Path Intellisense
- vscode-icons



기본 실습 과정

1. VS Code에서 새로운 파일 생성
   : 일차별로 바탕화면에 폴더 생성 '1day'

2. Live Server로 실행 → 크롬 브라우저에서 실행 확인

3. 브라우저 개발도구 Open (Ctrl+Shift+i) → 콘솔 탭

* 매일 수업을 마치면 실습 파일 압축해서 이메일로 제출
   : topmentor@daum.net
   : [이름] 1일차 실습파일

_Javascript에 대해

자바스크립트(JavaScript)란?

자바스크립트(JavaScript)는 **객체(object) 기반의 스크립트 언어**
자바스크립트는 1995년에 넷스케이프(Netscape)의 브렌던 아이크(Brendan Eich)에 의해 만들어짐.
처음에는 모카(Mocha)라는 이름으로 개발되었으나, 그 후에 라이브스크립트(LiveScript),
최종적으로 자바스크립트(JavaScript)라는 이름으로 변경.

자바스크립트의 특징

1. 자바스크립트는 **객체 기반의 스크립트 언어**
2. 자바스크립트는 동적이며, 타입을 명시할 필요가 없는 **인터프리터 언어**
3. 자바스크립트는 **객체 지향형 프로그래밍과 함수형 프로그래밍을 모두 표현**할 수 있음

자바스크립트 표준

1996년에 넷스케이프(Netscape)는 자바스크립트를 국제 표준안으로 만들기 위해 ECMA(European Computer Manufacturers Association)에 제출합니다. 그 결과 ECMA는 ECMAScript라는 새로운 표준을 제정하였고, 그 첫 번째 버전인 ECMA-262를 1997년에 공표합니다.
ECMAScript는 자바스크립트뿐만 아니라 마이크로소프트의 JScript나 어도비의 액션스크립트도 따르는 국제 표준이 됩니다. 현재 자바스크립트의 최신 표준은 2015년에 발표된 ECMAScript 6입니다.

_기본 입력과 출력

1. 기본 출력 : console.log(“출력 내용”);
   document.write("출력 내용");
2. 기본 입력 : prompt(“메시지”, 디폴트값) ;

```
var num = 10;
console.log("num 값은 ", num);
console.log("num 값은 " + num);
console.log(`num 값은 ${num}` );

document.write("num 값은 ", num, "<br>");
document.write(`num 값은 ${num}`, "<br>");

num = prompt("새로운 값 입력", 0);
console.log("num 값은 ", num);
```

이스케이프 문자	설명
\t	수평 탭
\n	줄바꿈
\'	작은따옴표
\"	큰따옴표
\\	역슬래시

_자바스크립트 변수

- 변수 선언 : let [변수명] = [초기값] ; var [변수명] = [초기값] ;
- 상수 선언 : const [상수명] = [초기값] ;

```
let num1 = 10;
let num2 = 20;
console.log(`num1 값은 ${num1}  num2 값은 ${num2}`);
console.log("두 값의 합은 " , num1 + num2 );

const num3 = 100;
console.log(`num3 값은 ${num3}` );
num3 = 200;
console.log(`num3 값은 ${num3}` );
```

```
▶Uncaught TypeError: Assignment to constant variable.
    at ex2.html:18:10
```

_자바스크립트 기본 자료형

- 숫자 : 정수, 실수 (타입으로 구분하지 않지만 변환시에는 구분함)
- 문자열 : " "로 묶음
- 템플릿 문자열 : ` `로 묶고 ${ }안에 변수나 간단한 연산을 넣을 수 있음
- 불리언 : true/false
  . undefined 자료형 : 초기화 하지 않음 변수
  . NaN : 숫자의 형식이지만 숫자 타입이 아닌 것
- typeof 연산자 : 변수의 타입을 알려 줌

이스케이프 문자	설명
\t	수평 탭
\n	줄바꿈
\'	작은따옴표
\"	큰따옴표
\\	역슬래시

```
let value;
let value1 = 10;
let value2 = "10";
let value3 = true;
let value4 = {};
let value5 = [];

console.log(`value의 타입은 ${typeof value} ` );
console.log(`value1의 타입은 ${typeof value1} ` );
console.log(`value2의 타입은 ${typeof value2} ` );
console.log(`value3의 타입은 ${typeof value3} ` );
console.log(`value4의 타입은 ${typeof value4} ` );
console.log(`value5의 타입은 ${typeof value5} ` );
```

_자바스크립트 자료형 변환

- 숫자 → 문자열 : String(숫자) , 변수.toString() , (숫자).toString()
- 문자열 → 숫자 : Number("문자열") , parseInt("문자열") , parseFloat("문자열")
- 숫자 + 문자열 = 문자열

  * 다른 연산자는 문자열이 숫자로 변환
- !! : 불리언 타입으로 변환
- ===, !== : 값은 물론 타입까지 같은 지 확인

```
let value1 = 10;
let value2 = "20";
let value3 = String(value1);
let value4 = Number(value2);
// value3 = value1.toString();
// value4 = parseInt(value2);
console.log(`value3 = ${value3} 의 타입은 ${typeof value3} ` );
console.log(`value4 = ${value4} 의 타입은 ${typeof value4} ` );
console.log(`value = ${value4} 의 불타입은 ${!!value4} ` );
console.log(`value = ${value1} + ${value2} 의 불타입은 ${value1 + value2} ` );
console.log(`50 == "50"의 결과는 ${50 == "50"} ` );
console.log(`50 === "50"의 결과는 ${50 === "50"} ` );
```

_자바스크립트 연산자

- 연산자 순위 : 증감 > 사칙/나머지 > 비교 / 일치 > 논리 > 대입

표 2-4 기본적인 사칙 연산자

연산자	설명
+	덧셈 연산자
-	뺄셈 연산자
*	곱셈 연산자
/	나눗셈 연산자

표 2-5 나머지 연산자

연산자	설명
%	나머지 연산자

표 2-13 숫자에 적용하는 복합 대입 연산자

연산자	설명
+=	숫자 덧셈 후 대입 연산자
-=	숫자 뺄셈 후 대입 연산자
*=	숫자 곱셈 후 대입 연산자
/=	숫자 나눗셈 후 대입 연산자

표 2-15 증감 연산자

연산자	설명
변수++	기존 변수 값에 1을 더합니다(후위).
++변수	기존 변수 값에 1을 더합니다(전위).
변수--	기존 변수 값에서 1을 뺍니다(후위).
--변수	기존 변수 값에서 1을 뺍니다(전위).

표 2-9 비교 연산자

연산자	설명
==	같습니다.
!=	다릅니다.
>	왼쪽 피연산자가 큽니다.
<	오른쪽 피연산자가 큽니다.
>=	왼쪽 피연산자가 크거나 같습니다.
<=	오른쪽 피연산자가 크거나 같습니다.

표 2-18 일치 연산자

연산자	설명
===	자료형과 값이 같은지 비교합니다.
!==	자료형과 값이 다른지 비교합니다.

표 2-10 논리 연산자

연산자	설명
!	논리 부정 연산자
||	논리합 연산자
&&	논리곱 연산자

_자바스크립트 연산자

```
let input = 0;
input = prompt("숫자를 입력하시오", 0);

let result = input + 20;
console.log(`input + 20 = ${result}`);

result = input % 2;
console.log(`input % 2 = ${result}`);

result += 100;
console.log(`result += 100 -- ${result}`);

result++;
console.log(`result++  -- ${result}`);
```

_조건문 - if

- 기본형 if문
- 중첩형 if문

```
if (불_표현식) {

}
```

```
if (불_표현식) {
    // 불_표현식이 참일 때 실행할 문장
} else {
    // 불_표현식이 거짓일 때 실행할 문장
}
```

```
if (불_표현식) {

} else if (불_표현식) {

} else if (불_표현식) {

} else {

}
```

```
if (불_표현식) {
    if (불_표현식) {
        문장;
    } else {
        문장;
    }
} else {
    if (불_표현식) {
        문장;
    } else {
        문장;
    }
}
```

_조건문 - if

```
let input = 0;
input = prompt("숫자를 입력하시오", 0);

if (input % 2 == 1) {
      console.log(`${input}은 홀수`);
} else {
      console.log(`${input}은 짝수`);
}

if (input % 2 == 0 || input % 3 == 0) {
      console.log(`${input}은 2의 배수 이거나 3의 배수`);
}
```

_반복문 - while 반복문

- while문 : 조건이 참인 동안 반복

```
while (불_표현식) {
    // 불 표현식이 참인 동안 실행할 문장
}
```

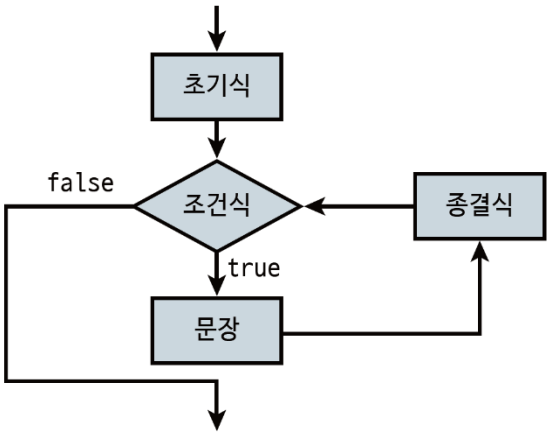
```
let i = 0;
let num = 10;

while (i < num) {
    console.log(i);
    i++;
}
```

```
let num = 0;

while (true) {
    num = prompt("숫자를 입력하시오", 0);
    console.log(`input num = ${num}`);
    if (num == 0) {
        console.log(`종료`);
        break;
    }
}
```

_반복문 - for 반복문

- for문 : 초기식 + 조건식 + 종결식(증감식)

```
for (let i = 0; i < 반복_횟수; i++) {

}
```




```
let output = 0;

for (let i = 0; i <= 100; i++) {
    output += i;
}

console.log(output);
```

_반복문 - for-in / for-of 객체 순회 반복문

- 배열이나 객체(Collection)의 요소를 순회하는 반복문
- for-in : 순회하면서 인덱스를 알려 줌
- for-of : 순회하면서 요소를 알려 줌

```
for(let 인덱스 in 배열) {

}
```

```
for(let 요소 of 배열) {

}
```

```
let arr = ["사과", "배", "복숭아", "귤"];
for (let i in arr) {
   console.log(i);
}

for (let i of arr) {
   console.log(i);
}
```

_내장 객체

- Date 객체

클래스	설명
new Date()	현재 시간으로 Date 객체를 생성합니다.
new Date(유닉스_타임)	유닉스 타임(1970년 1월 1일 00시 00분 00초부터 경과한 밀리초)으로 Date 객체를 생성합니다.
new Date(시간_문자열)	문자열로 Date 객체를 생성합니다.
new Date(연, 월 - 1, 일, 시간, 분, 초, 밀리초)	시간 요소(연, 월 – 1, 일, 시간, 분, 초, 밀리초)를 기반으로 Date 객체를 생성합니다.

*** Month를 나타내는 '월'은 0부터 시작 (0 = 1월, 11 = 12월)**

```
// 현재 시간을 기반으로 Date 객체를 생성합니다.
let dateA = new Date();
console.log(dateA);
// 유닉스 타임(1970년 1월 1일 00시 00분 00초부터 경과한 밀리초(ms)를 정수 형태로 나타냅니다.)
let dateB = new Date(692281800000);
console.log(dateB);

// 문자열을 기반으로 Date 객체를 생성합니다.
let dateC = new Date("December 9, 1991 21:30:00")
console.log(dateC);

// 시간 요소(연, 월 - 1, 일, 시간, 분, 초, 밀리초)를 기반으로 Date 객체를 생성합니다.
let dateD = new Date(1991, 12 - 1, 9, 21, 30, 0, 0);
console.log(dateD);
```

_내장 객체

- Math 객체 : 자주 사용하는 수학 함수와 상수

1. Math.min()
```
Math.min(1, 10, -100, -10, 1000, 0);    // -100
```

2. Math.max()
```
Math.max(1, 10, -100, -10, 100, 0);
```

3. Math.random() : 0~1사이 임의 값
```
Math.floor(Math.random() * 10);  // 0~9 사이 정수
Math.floor(Math.random() * 100);  // 0~99 사이 정수
Math.random() * (max - min) + min;    // min~max 사이 임의 실수
Math.floor(Math.random() * (max - min) + min);    // min~max 사이 임의 정수
```

4. Math.round() : 소수 첫째자리 반올림
```
Math.round(10.49);  // 10
```

5. Math.floor() : 버림

6. Math.ceil() : 올림

_함수 - 기본정의

- 함수 = 모듈
- 정의형식 : function [함수명] (가인자:인자정의) { 함수 로직 }
- 호출형식 : [함수명](실인자:실재값);
- 리턴할 값이 있으면 return 문 뒤에 명시

```
let num = 10;
function plusNum(value) {
  console.log("value의 값은 " + value + "입니다.");
  let num = value + 10;
  return num;
}
console.log("함수 호출 결과 값은 " + plusNum(num) + "입니다.");
```

_문자열

- 문자열은 기본적으로 상수의 속성임
- 길이 : length
- 문자단위 접근 : charAt([인덱스]), charCodeAt([인덱스]) → 유니코드 값

```
var firstStr = "이것도 문자열입니다."; // 큰따옴표를 사용한 문자열
var secondStr = "이것도 문자열입니다."; // 작은따옴표를 사용한 문자열
var thirdStr = "나의 이름은 '홍길동'이야."; // 작은따옴표는 큰따옴표로 둘러싸인 문자열에만 포함
var fourthStr = '나의 이름은 "홍길동"이야.'; // 큰따옴표는 작은따옴표로 둘러싸인 문자열에만 포함

var strKor = "한글";
var strEng = "abcABC";
console.log(strKor.length); // 2
console.log(strEng.length); // 6

str = "abcDEFabc";
console.log(str.charAt(0)); // a
console.log(str.charCodeAt(0)); // 97        -> 'a'에 해당하는 UTF-16 코드를 반환함.
```

_문자열 - 찾기

- indexOf([찾을 문자열], [시작인덱스:생략하면 0부터]); → 찾은 시작 인덱스를 알려 줌 (못 찾으면 -1)
- lastIndexOf([찾을 문자열], [시작인덱스:생략하면 맨뒤부터]); → 찾은 시작 인덱스를 알려 줌 (못 찾으면 -1)
- search([찾을 문자열]); → 찾은 시작 인덱스를 알려 줌 (못 찾으면 -1)
- replace(/[찾을 문자열]/gi, [바꿀문자열]) → 문자열을 찾아 바꿈
- replaceAll([찾을 문자열], [바꿀문자열]) → 문자열을 찾아 바꿈

```
let str = "abcDEFabc";
console.log(str.indexOf("abc")); // 0  -> 자바스크립트에서 인덱스는 0부터 시작함.
console.log(str.search("abc")); // 0  -> 자바스크립트에서 인덱스는 0부터 시작함.
console.log(str.indexOf("abcd")); // -1 -> 문자열을 비교할 때 문자의 대소문자를 구분함.
console.log(str.indexOf("abc", 3)); // 6  -> 인덱스 3부터 'abc'를 찾기 시작함.
console.log(str.lastIndexOf("abc")); // 6
console.log(str.lastIndexOf("d")); // -1
console.log(str.lastIndexOf("c")); // 8

console.log(str.replace("abc", "ddd"));  // 1개만 바꿈
console.log(str.replaceAll("abc", "ddd"));  // 전체 바꿈
```

_문자열 - 문자열 추출

- 인덱스로 지정 추출 : slice([시작인덱스], [끝인덱스]) , substring([시작인덱스], [끝인덱스])
- 인덱스와 개수로 지정 : substr([시작인덱스], [추출문자수])
- 구분자로 문자열 나누기 : split([구분자]) → String 배열로 리턴

```
str = "abcDEFabc";
console.log(str.slice(2, 6)); // cDEF     -> 인덱스 2부터 인덱스 5까지의 문자열을 추출함.
console.log(str.slice(0, -1)); // abcDEFab -> 마지막 문자 제거
console.log(str.substring(2, 6)); // cDEF   -> 시작인덱스, 끝인덱스
console.log(str.substr(2, 4)); // cDEF      -> 시작인덱스, 추출문자수

let str2 = "자바스크립트는 너무 멋져요! 그리고 유용해요.";
console.log(str2.split("")); // 한 문자("")씩 나눔.
console.log(str2.split(" ")); // 띄어쓰기(" ")를 기준으로 나눔.
console.log(str2.split("!")); // 느낌표("!")를 기준으로 나눔.
```

_문자열 - 기타

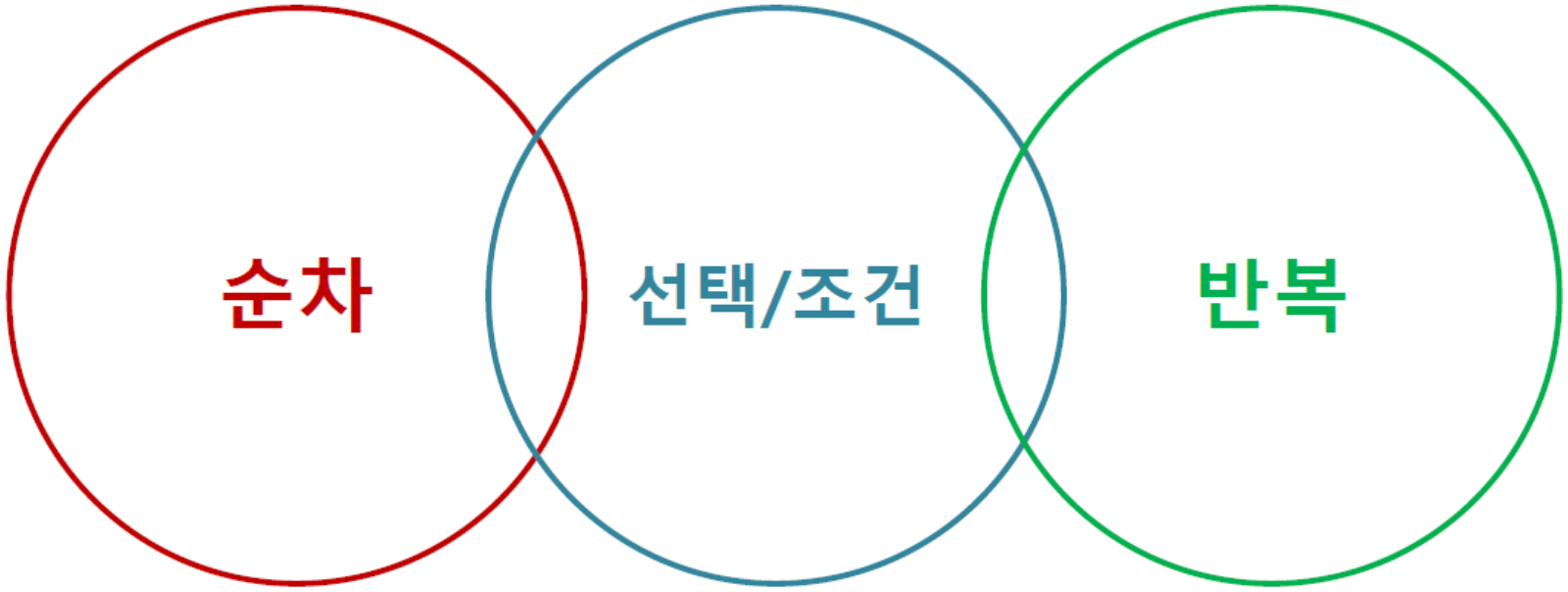
- 문자열 합치기 : concat([추가문자열])
- 대문자, 소문자 변경 : toUpperCase(), toLowerCase()
- 양쪽 빈공백 제거 : trim()

```
let str3 = "자바스크립트";
console.log(str3.concat("는 너무 멋져요!")); // 자바스크립트는 너무 멋져요!

str = "abcDEFabc";
console.log(str.toUpperCase());
console.log(str.toLowerCase());

let str4 = "          자바스크립트          ";
console.log(str4.trim());
```

응용 예제

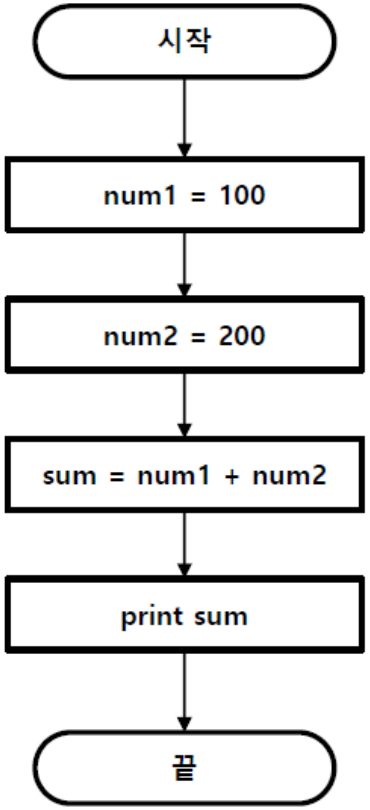
프로그램 논리

프로그램 작성에 필요한 기본 논리는 순차 , 선택 , 반복 3 가지 논리로 구성된다

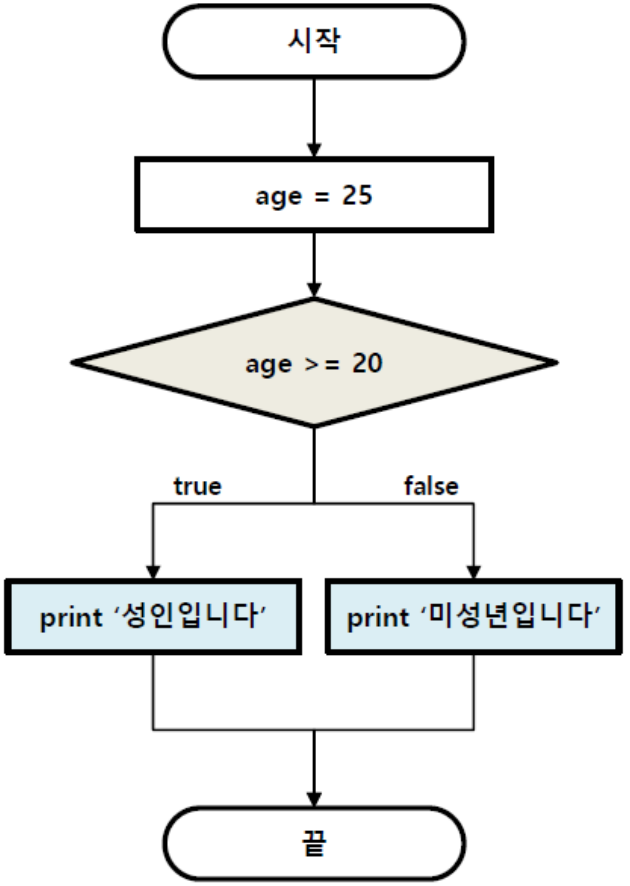


프로그램 논리

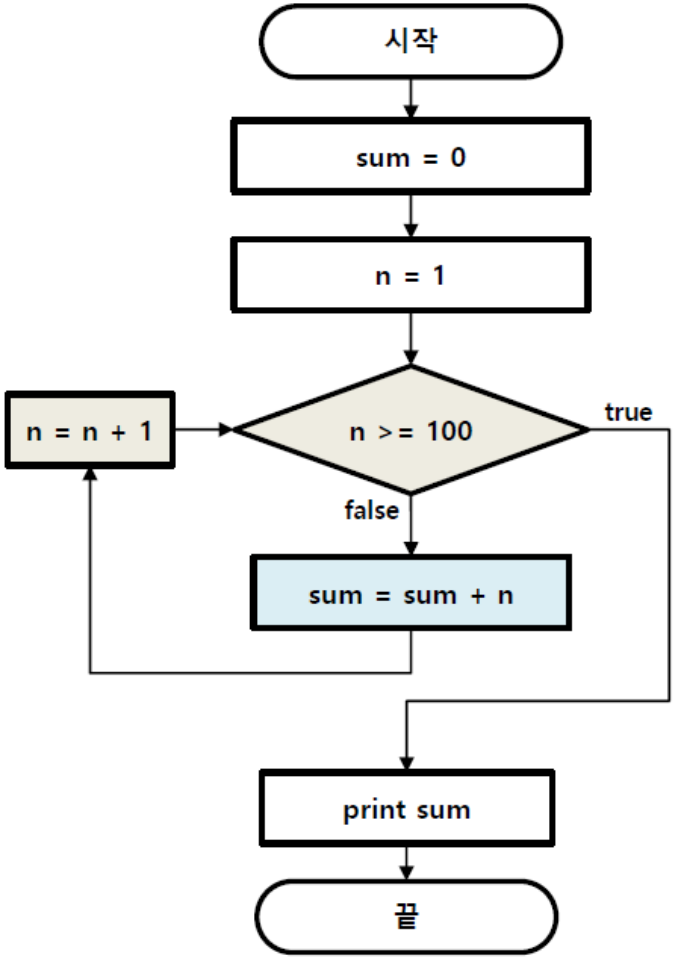
순차



조건

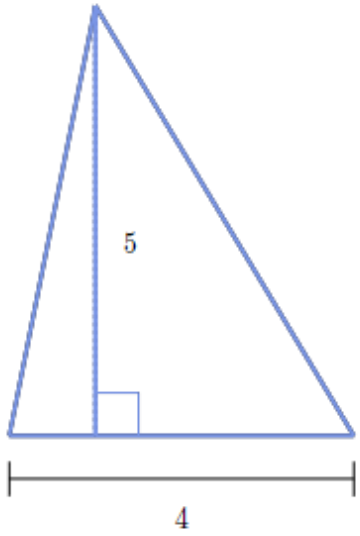


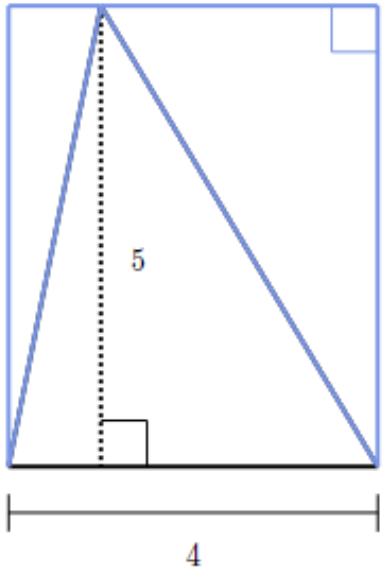
반복



2_실습-1

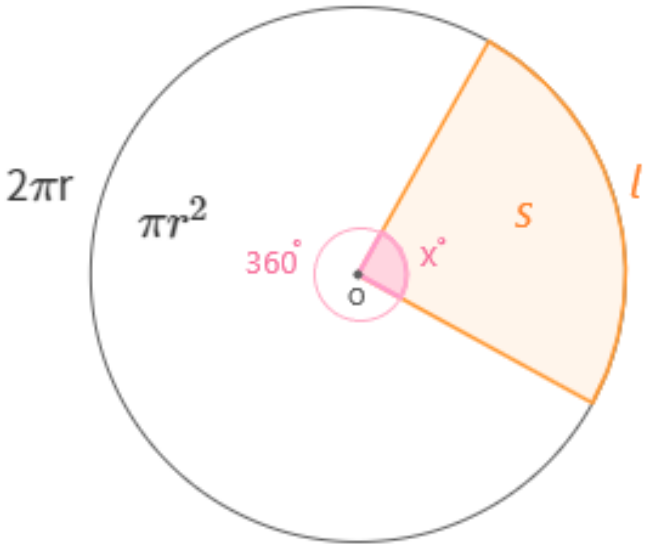
실습 문제 : 높이와 너비를 입력 받아서 삼각형의 넓이를 출력하는 프로그램을 작성하시오





2_실습-2

실습 문제 : 원의 반지름을 입력 받아 원의 둘레와 면적을 출력하는 프로그램을 작성하시오.
(원주율은 3.14로 함)



실습-3

실습 문제 : 1~100까지 3의 배수만 누적해서 더한 결과를 출력하는 프로그램

실습-4

실습 문제 : 아래와 같이 별 피라미드가 출력되는 프로그램

```
*
**
***
****
*****
******
*******
********
*********
**********
```

실습-4

실습 문제 : 아래와 같이 별 피라미드가 출력되는 프로그램

```
         *
        **
       ***
      ****
     *****
    ******
   *******
  ********
 *********
**********
```