

2021/10/8 (exam 1/test 3)

* 배열 → JSON 문자열.

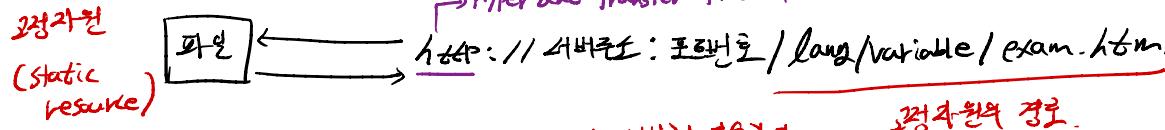
new String[] {"aaa", "bbb", "ccc"} → ["aaa", "bbb", "ccc"]

new int[] {100, 200, 300} → [100, 200, 300]

* 자원과 그자리를 가드하는 주소

Resource Unified Resource Locator = URL

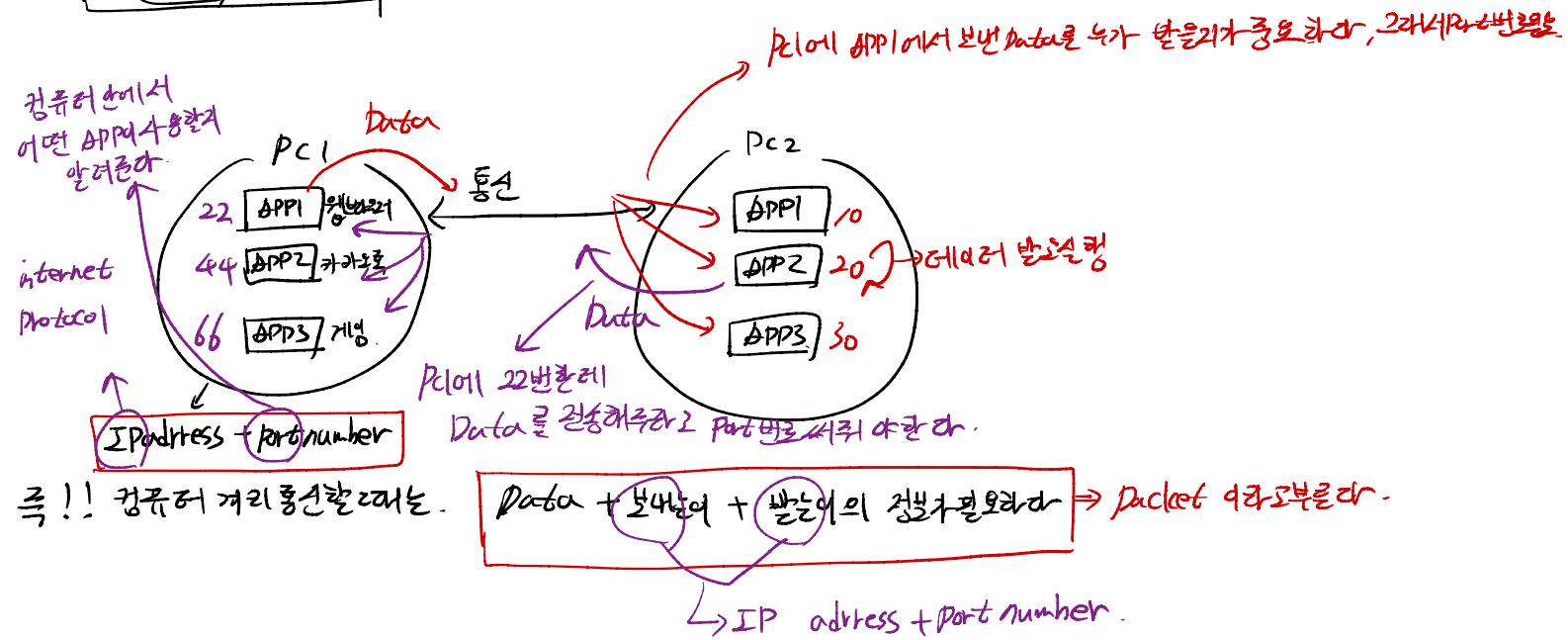
→ HyperText Transfer Protocol



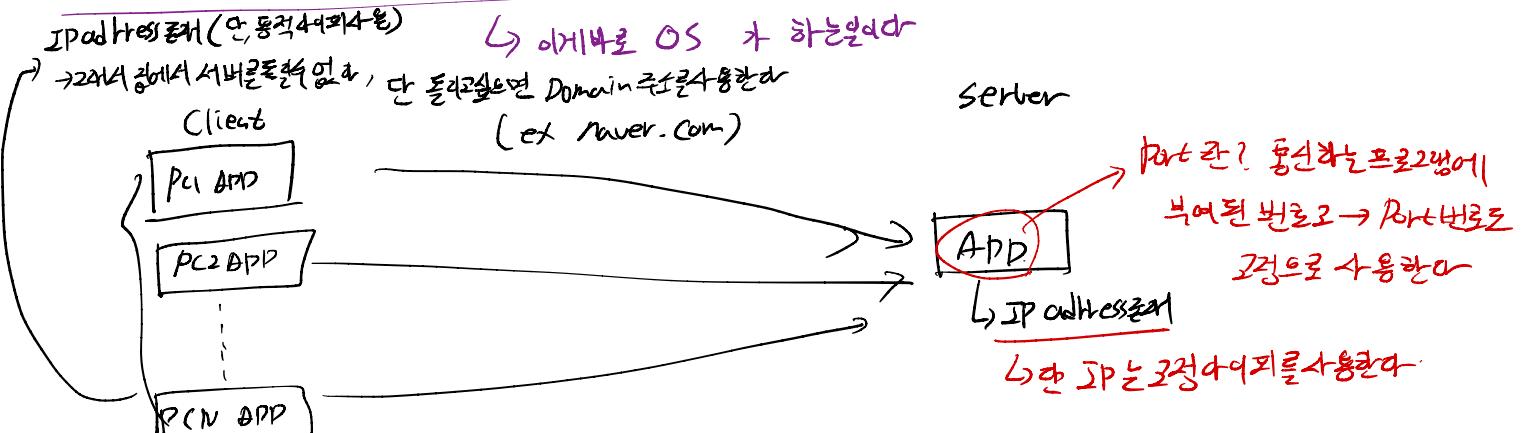
정적자원의 경로.



여기로 들어오는게 오는가, 그래서 Port번호가 필요하다.

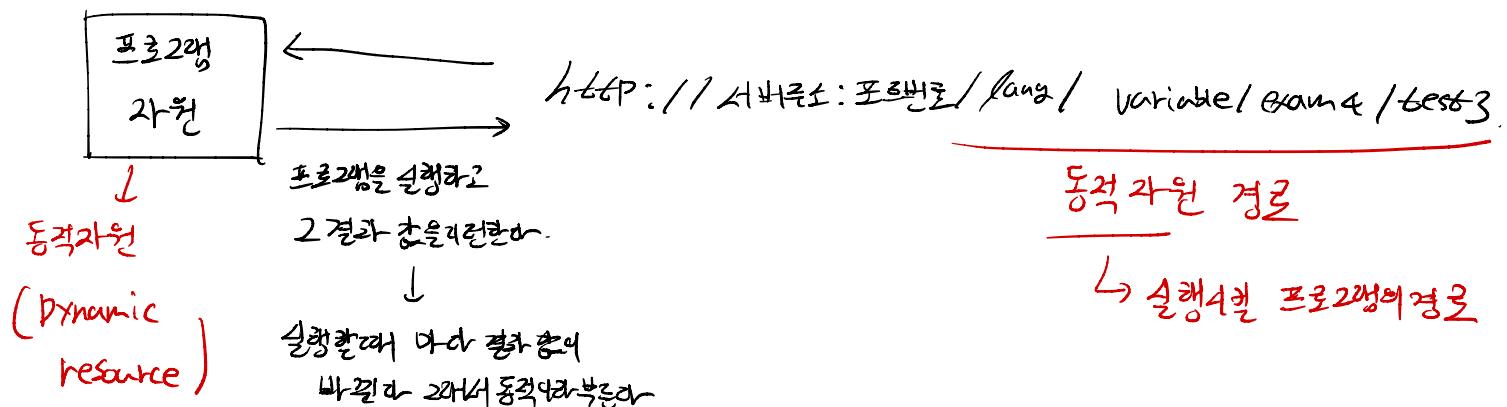


ex)



*같은 서버 프로그램은 port 번호가 고정되어 있다 (IP주소가 달라도 port 번호 같다)

※) Web Server : port = 80 , https = 443



* (Exam4 / test 3)

- 콜백함수: 내가 작업호출을 래서 사용하는 게 아니고 다른 명령문에 호출하는 것이다. ex) `foreach(names, cb)`

↳ for 문이 돌 때마다
for가 callback함수를
호출한다.

* function chain

$f_1()$ 실행

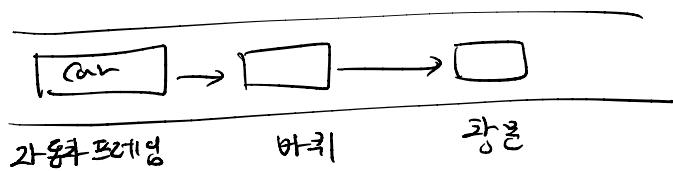
↳ 결과에 대해 $\rightarrow f_2()$ 실행

↓ 같다

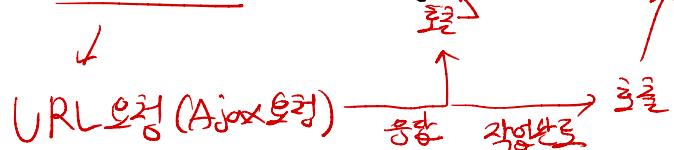
↳ 결과에 대해 $\rightarrow f_2()$ 실행

$\Rightarrow f_1(), f_2(), f_2()$

ex) 자동차 ^{운전} 커넥티비티



ex) fetch().then(cb).then(cb)



* 연산자와 데이터 타입 \rightarrow ① 같은 타입끼리만 연산가능, ② 다른 타입인 경우 형변환 후 연산수행

$$\begin{array}{c} \text{byte} + \text{byte} \\ \downarrow \quad \downarrow \\ \text{int} \quad \text{int} \quad \text{int} \Rightarrow \text{int} \end{array}$$

$$\begin{array}{c} \text{short} + \text{short} \\ \downarrow \quad \downarrow \quad \downarrow \\ \text{int} \quad \text{int} \quad \text{int} \Rightarrow \text{int} \end{array}$$

$$\begin{array}{c} \text{short} + \text{byte} \\ \downarrow \quad \downarrow \quad \downarrow \\ \text{int} \quad \text{int} \quad \text{int} \Rightarrow \text{int} \end{array}$$

$$\begin{array}{c} \text{int} + \text{long} \\ \downarrow \quad \downarrow \\ \text{int} \quad \text{int} \quad \text{long} \Rightarrow \text{long} \\ \downarrow \\ \text{long} \end{array}$$

$$\begin{array}{c} \text{int} + \text{float} \\ \downarrow \quad \downarrow \\ \text{float} \quad \text{float} \Rightarrow \text{float} \end{array}$$

$$\begin{array}{c} \text{long} + \text{float} \\ \downarrow \\ \text{float} \end{array} \quad \begin{array}{c} \text{float} \\ \downarrow \\ \text{+} \\ \Rightarrow \text{float} \end{array}$$

* 양수적 형변환 VS

영수적 형변환

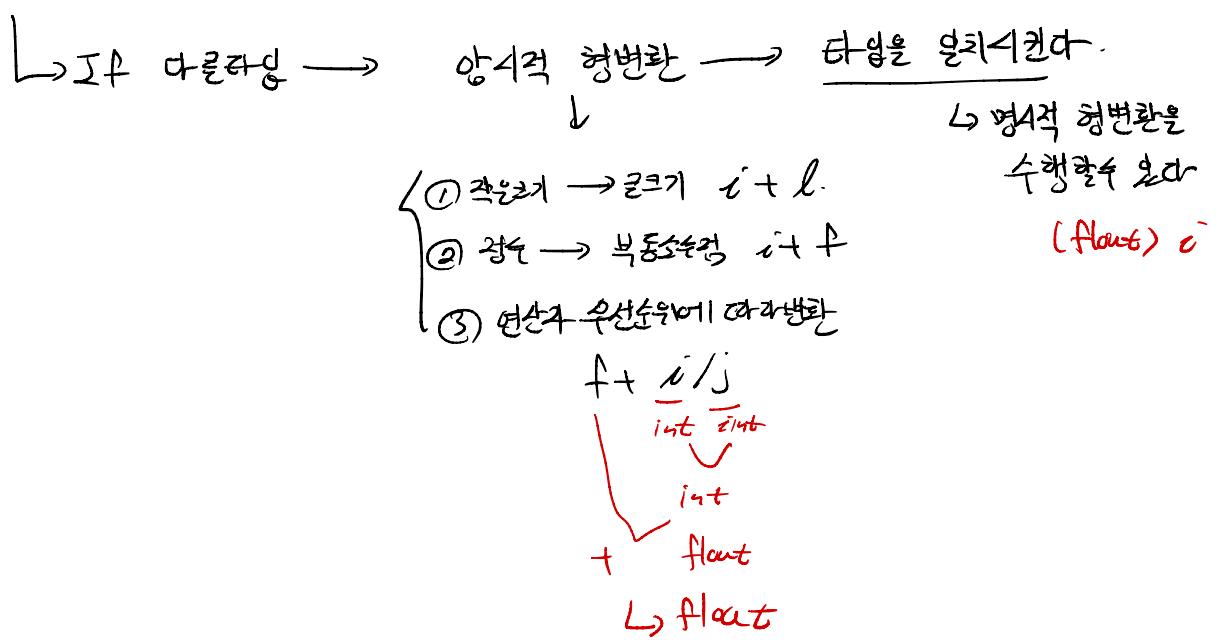
int $i = 100$
float $j = 3.14f$,
 $\text{float } s = i + j$,
양수적
형변환
↓
[100.0]
4376c
+ ⇒ float
입력 float 예상되잖아.

int $i = 5$,
int $j = 2$,
float s ,
 $s = \frac{(\text{float}) i}{(\text{float}) j}$,
5.0 / 2.0
Float 2.5.

* 산술연산자 정리 (+, -, *, /, %)

① 정수 기본 연산은 int이다

② 같은 타입끼리만 연산가능 \Rightarrow 결과도 같은 타입



③ 연산자 우선순위

(i) ()

(ii) ++a, a++

(iii) *, /, %

(iv) +, -

* 비교연산자 ($<$, \leq , $>$, \geq , $=$, \neq) 정리

① 부동소수점 비교 예측 결과와 다른 경우 조심!

float $f1 = 12.321f1;$

float $f2 = 1.1f1;$

$$+ \begin{cases} f1 = 12.321 \\ f2 = 1.1 \end{cases}$$

$13.421 \Rightarrow$ 예측값
#

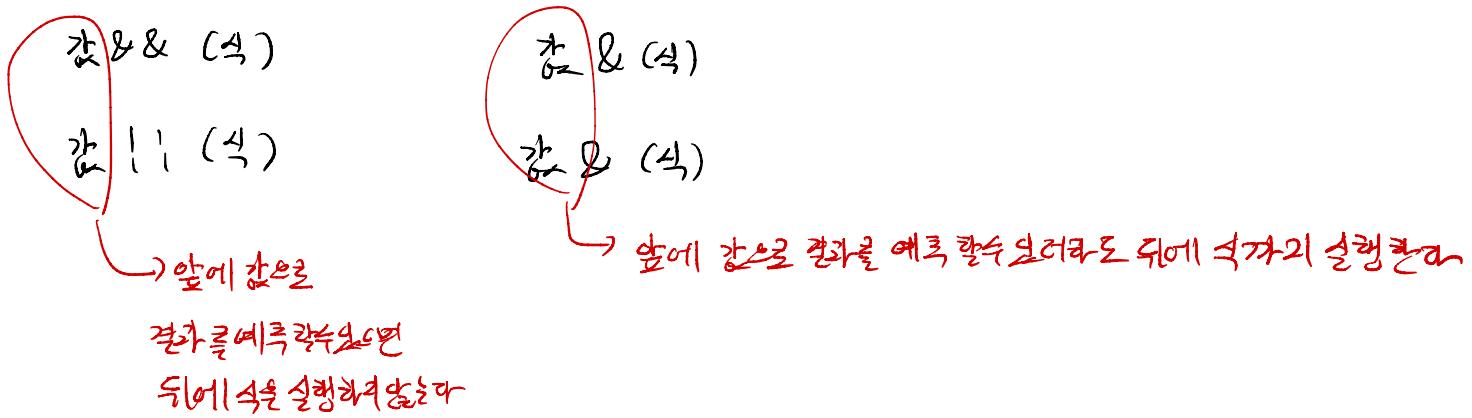
$13.421005 \Rightarrow$ 결과값

↳ IEEE754에서
부동소수점을 2진수로 바꿀 때
크소수의 값이 블여준다.

*논리 연산자 (&&, ||, ^, !)
and or exclusive or not

- ① boolean 타입에 대처 사용
- | | |
|--------------------|-----------------------|
| boolean && boolean | → 둘다 true일 경우 true |
| boolean boolean | → 둘중 하나 true일 경우 true |
| boolean ^ boolean | → 두자의 값이 다른 경우 true |
| ! boolean | → 반대로 |

② && vs &, || vs |



* 바로 놀리 연산자 (&, !, ^, ~) 전치
and or exclusive not

① 4종 { 정수 & 정수
정수 ! 정수
정수 ^ 정수
~정수 } → 바로 한위로 실행.

② 활용 { & → 기준식상 빼면, 좌연산자로 하면 → % 연산 수행 (단, 2의제곱수를 빼거나)
! → 기준식상 빼면,
^ → 그대로 기법
~ → 색상 반전

* 비트 이동 연산자

① 사용

정수 << 이동비트 정수 >> 이동비트 정수 >>> 이동비트	\Rightarrow 정수 $\times 2^{\text{이동비트수}}$ \Rightarrow 정수 $\times 2^{-\text{이동비트수}}$
---	---

② 활용

$$\begin{array}{ll} 21 \times 2 = 21 \ll 1 & 21 / 2 = 21 \gg 1 \\ 21 \times 4 = 21 \ll 2 & 21 / 4 = 21 \gg 2 \\ 21 \times 8 = 21 \ll 3 & 21 / 8 = 21 \gg 3 \end{array}$$

↳ 산술연산보다 속도가 빠르다

정수 << 비트이동 개수.

$$\begin{array}{c} \uparrow \\ \text{int} \end{array} \quad \hookrightarrow \frac{\text{비트이동개수 \% } 32}{11} = 0 \sim 31 \rightarrow \text{이동개수 유로 범위}. \\ \text{비트이동개수 \% } 32 = 00011111$$

$$\begin{array}{r} 8 \quad 6 / 00001 \rightarrow 33 \% 32 = 1 \\ \hline 001111 \\ \hline 000001 \rightarrow 1 \end{array}$$

$$0 / 000001 \rightarrow 60 \% 32 = 1$$

$$94 \% 32 = 1$$

* 비트 이동 연산 적용.

① 특정 위치의 바이트 값을 추출할 때

a	21	a1	30	ff
	21	10000000	10011000	11111111

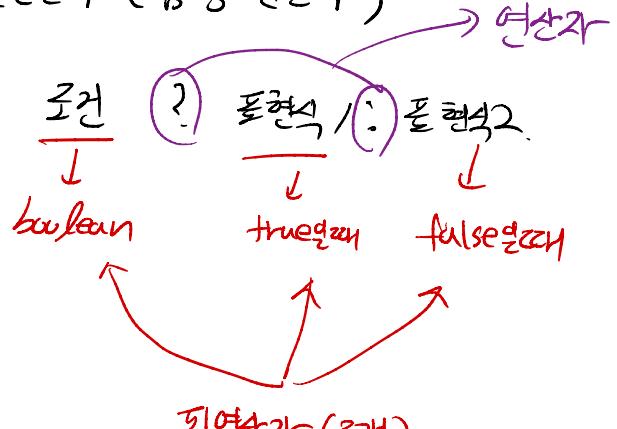
$$a \gg 24 = \boxed{00\ 00\ 00\ 21} \quad a1\ 30\ ff$$

$$a \gg 16 = \boxed{00\ 00\ 21\ a1} \quad 30\ ff$$

& 000000 ff

0 0 0 0 0 a1

* 조건연산자 (삼항 연산자)



* 문장 VS 표현식

실행명령

System.out.println

`int a=100;`

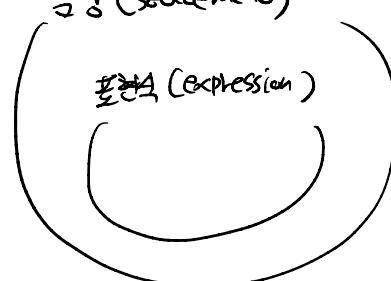
`if (a>22) { }`

`a+2` → 정수자료형

`Math.abs(-200)` → 정수자료형

표현식

문장



66Hello²², charAt(2) → 문자자료형