**RECORD**

우선 자바스크립트의 기본적인 ‘문’처리 방식을 배울 필요가 있다.  
자바스크립트는 ‘문’을 해석할 때 ‘식’하고는 다르게 해석한다.  
‘식’은 하나의 ‘값’으로 처리될 뿐이다.  
**그런데 ‘문’은 하나의 실행 단위로 해석이 된다.**  
그래서 우리가 문을 10개를 작성하면 10개의 실행 단위가 생긴다.  
그에 비해 ‘식’은 아무리 많이 써봐야 ‘값’으로 수렴할 뿐이다.  
자바스크립트는 우리가 ‘문’의 갯수를 늘리면 ‘문’의 갯수만큼 우리가 처리해야할 과제로 삼는다.  
따라서 한줄한줄 ‘문’이 생길 때마다 자바스크립트는 이를 ‘과제’로 등록한다.  
이 ‘과제’를 **RECORD**라고 부른다.

**Statement(문) - > Record(레코드)로 등록**

즉 자바스크립트는 문을 레코드로 등록하고 이 레코드를 하나하나 소비해나가며 과제를 해결한다.

**COMPLETION RECORD**

Flow control을 할 수 있는 제어문을 record로 바꿀 때는 Completion record라는 것으로 바뀐다.

Completion Record는 어떤 레코드를 선택할지를 결정하는데 사용된다.

**FLOW CONTROL (직접 흐름 제어)**

abc언어의 후손인지 알려면 플로우 컨트롤을 하기 위한 별도의 언어를 if, switch, for, while의 사용하는지 확인.

**DIRECT FLOW CONTROL (직접 흐름 제어)**

자바스크립트에서 사용할 수 있는 직접흐름제어 방법은 **Label**이다

**Label**

Identifier(식별자) : 자바스크립트 변수 이름 규칙과 한가지만 다르다. $를 LABEL 이름에 쓸 수 없다

**JS변수 이름 규칙:** 첫 글자가 알파벳, $, 언더바 그 이후의 문자는 알파벳, 숫자, $, 언더바

**Label의 이름 규칙:** JS변수 이름 규칙과 한가지만 다르다. $를 LABEL 이름에 쓸 수 없음

1. LABEL 외에 실제로 존재하는 ‘문’이 하나도 없으면 에러가 발생한다.
2. LABEL은 내부적으로 ‘문’이 아니라는 뜻이다.

3. LABEL은 브라우저 해석기가 볼 때는 ‘문’이 아니라는 것.

LABEL은 레코드에 달아주는 일종의 TAG(꼬리표) 같은 녀석인 것이다

1. LABEL은 같은 스코프 내에서 같은 이름을 쓰면 오류가 뜬다.

**LABEL의 SCOPE**

**Label scope 의 문법**

**label: {**

**}**

**Label Shadow**

**Static Parsing**

Label 스코프 내에서 label을 가지면 바깥 쪽의 이름을 가질 수 있다.

**LABEL RANGE & SET**

LABEL RANGE를 이해하려면 SET을 먼저 인식해야 한다.  
어떻게해서 LABEL이 LABEL 영역을 가지는지를 알아야된다는 것이다.  
위에선 LABEL SCOPE를 생성해주었지만 SCOPE를 생성해주지 않았을 때, 하나의 LABEL이 어디까지 영역을 잡느냐는 것이다.

1. LABEL에 특별한 무언가가 없다면 다음 LABEL까지를 범위로 삼는다.
2. LABEL 다음에 Iteration(for, while..)이 오거나 Switch가 오면 범위를 한정짓는다.  
   LABEL 다음에 for문이면 해당 LABEL의 범위는 그 for문까지~! 다음 LABEL이 없어도~!

break abc;를 했어도 다음으로 넘어가질 않는다. 위에서 LABEL RANGE는 특별한 경우가 없는 이상 다음 LABEL 까지라고 했다.  
하지만 break가 에러는 안 났지만 건너뛰진 않는다.

break로 건너뛸 수 있게 하려면 반드시 다음 3가지 구문 중 한 가지를 사용하여 LABEL을 써야된다.

1. Label Scope를 선언

2. Iteration Label이 되어라

3. Switch Label이 되어라

Auto Label(Undefined Named Label)

**Label을 Comment로 사용**

**Switch 문**

Special Label Block

Fall Through

If(식) 뒤에 내용이 옵셔널

If (식) else 필수적(멘다토리)