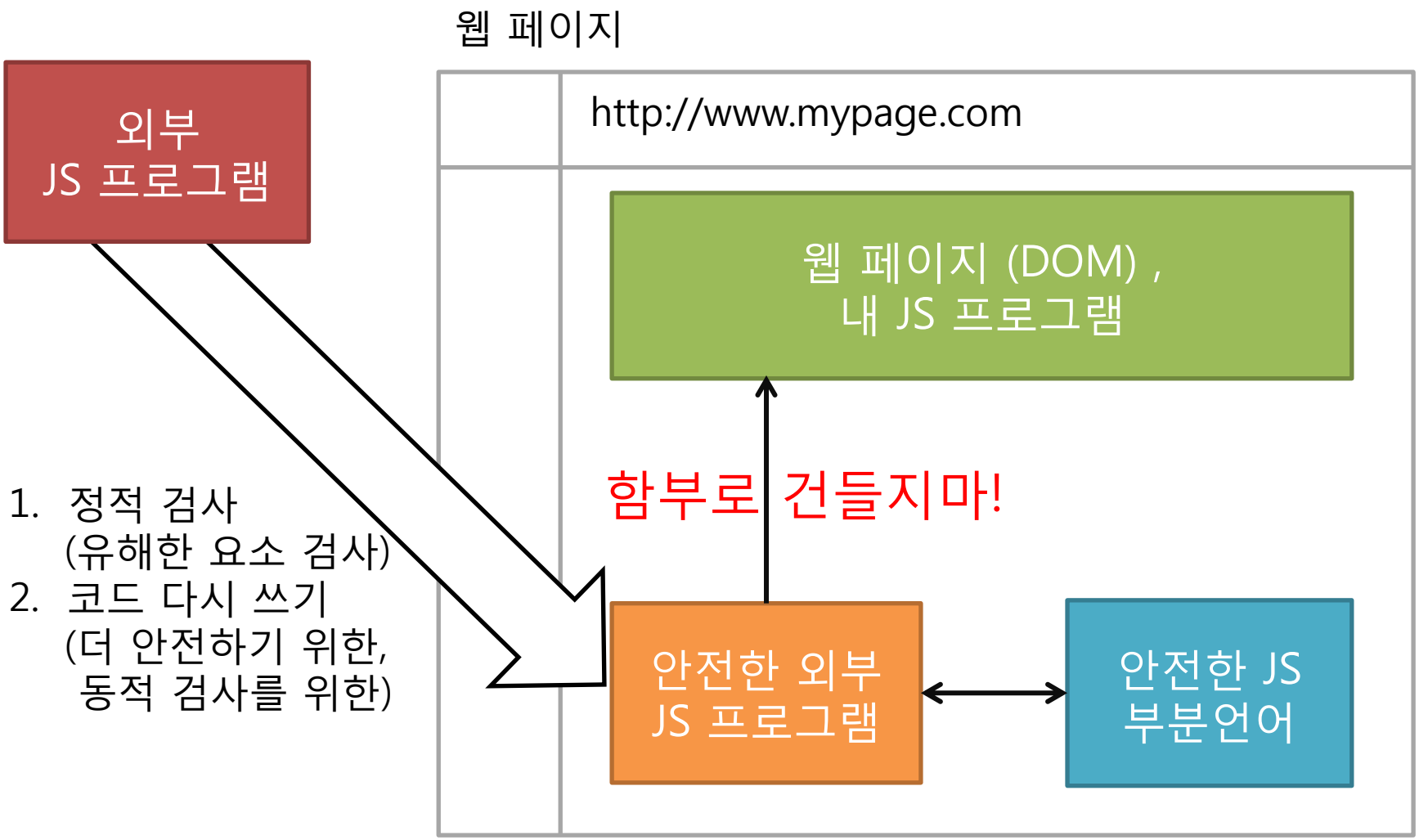
안전한 JavaScript 부분언어에 관한 조사

최재준
Programming Language Laboratory,
Computer Science Dept., KAIST

안전한 JavaScript 사용을 위한 부분언어들

- ADsafe
  - 안전한 웹 광고 제작
  - 제약이 많음, 코딩 패턴대로 사용
- Caja
  - 안전한 웹 어플리케이션 작성
  - JS 프로그램을 모듈형태로 변환, Cajita
- FBJS
  - 안전한 Facebook 웹 어플리케이션 작성
  - 비교적 적은 제약, Facebook에서만 사용

JavaScript 부분언어의 동작 방식



JavaScript 부분언어는 어떤 방식으로 안전하게 하나?

- 웹 어플리케이션이 허튼 짓을 못하게 한다

- 허튼 짓?
  - 외부 자원을 마음대로 접근 하거나 수정
  - 위험한 코드를 동적으로 실행

JavaScript 부분언어는 어떤 방식으로 안전하게 하나?

- 웹 어플리케이션을 고립시킨다
  "모래상자 안에서만 놀게 한다"
  - 전역 객체에 접근을 제한
  - DOM 접근 및 조작 제한
  - …
- 동적 요소를 줄인다
  - `eval()` 제거 (ADsafe, FBJS)
  - 안전한 `eval()` 제공(Caja)
  - …

JavaScript 부분언어가 못하게 하는 것 (1)

- this 키워드를 제한한다
  - this를 통해 전역 객체에 접근 가능

```
var obj = {
        "getThis": function() {return this} };
var f = obj.getThis;

> obj.getThis() → obj
> f() → window  // window는 전역객체


f().document. …      // DOM에 접근!
```

JavaScript 부분언어가 못하게 하는 것 (2)

- `eval()` 메소드 사용을 제한한다
  - 정적 검사를 힘들게 한다
  - `eval()`이 무슨 일을 할지 알 수 없다

    "Eval is evil"- Douglas Crockford

- `eval()`이 꼭 필요하면? (예 JSON)
  - `eval`과 유사한 안전한 메소드 제공 (Caja)
  - 안전한 JSON객체 제공 (Caja, FBJS)

JavaScript 부분언어가 못하게 하는 것 (3)

- 특정 객체속성에 접근을 제한한다.
  - `__proto__, prototype, arguments, callee, caller, constructor, eval, …`
- JavaScript에서 속성에 접근할 때는…

```
  obj.__proto__
≡ obj[“__proto__”]
≡ obj[“__pro” + “to__”]   // 정적 검사만으로는 불충분!
```

- 코드 다시 쓰기로 동적 검사 코드 삽입

(FBJS) `o[p] ⇒ o[idx(p)]`

`// idx(p) → bad, if p ∈ 접근금지목록`

앞으로 할 일

- 더 조사한다
- 어떻게 해야 정말 안전할까?
- 지금 충분히 안전할까?
- 얼마나 쓸만할까?
  - 표현력, 속도, 편한 사용