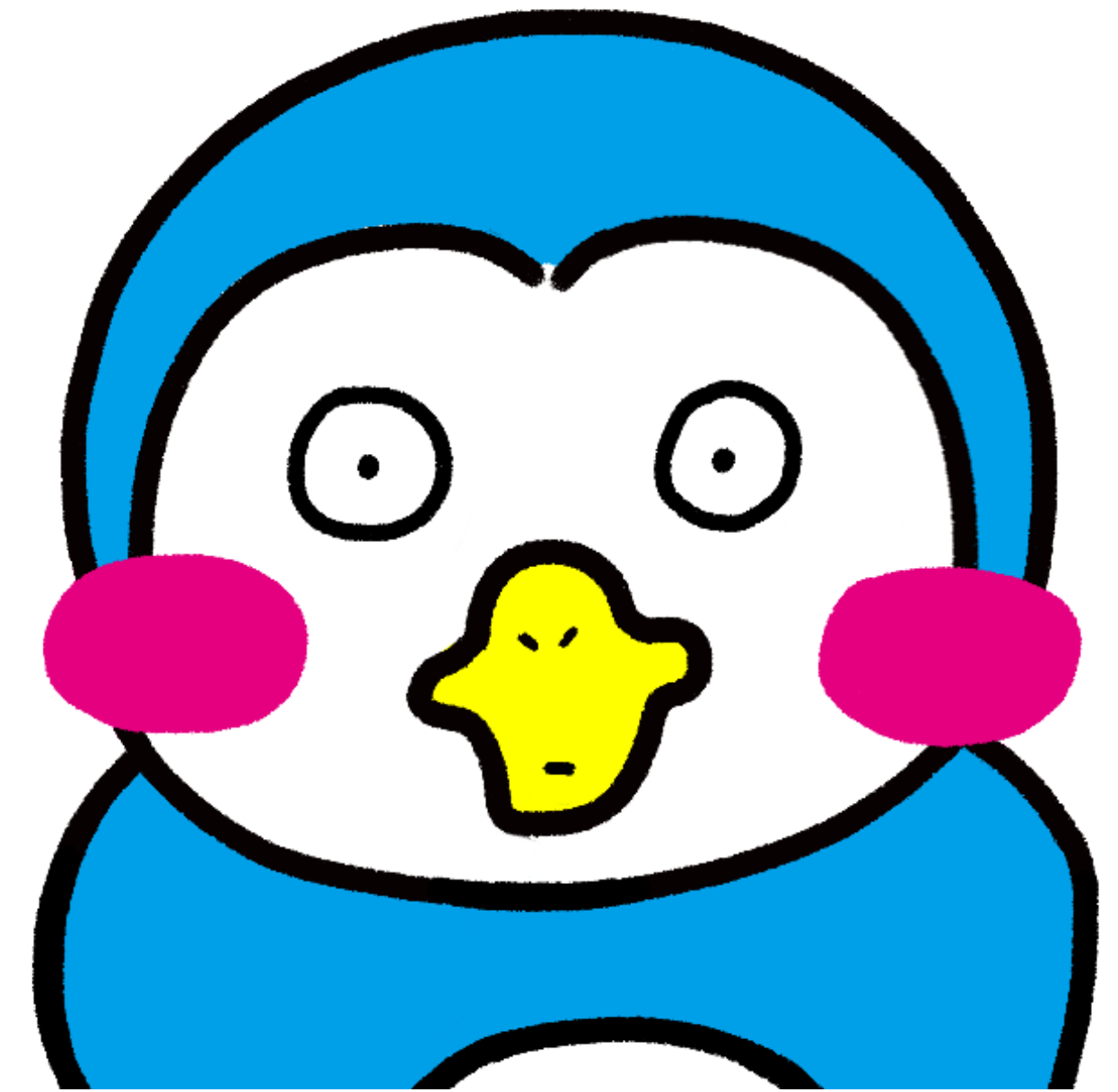


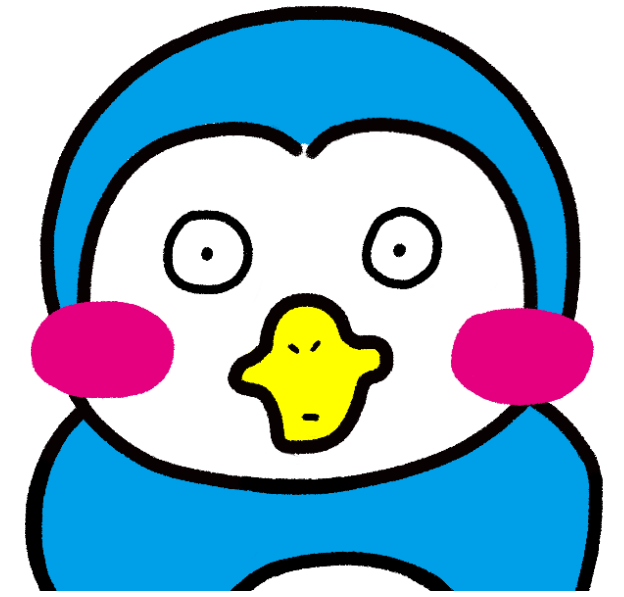
자 이제  
진짜(?) 시작이야



---

콘솔을 소개합니다 & 코드 작성 규칙

# 콘솔을 소개합니다

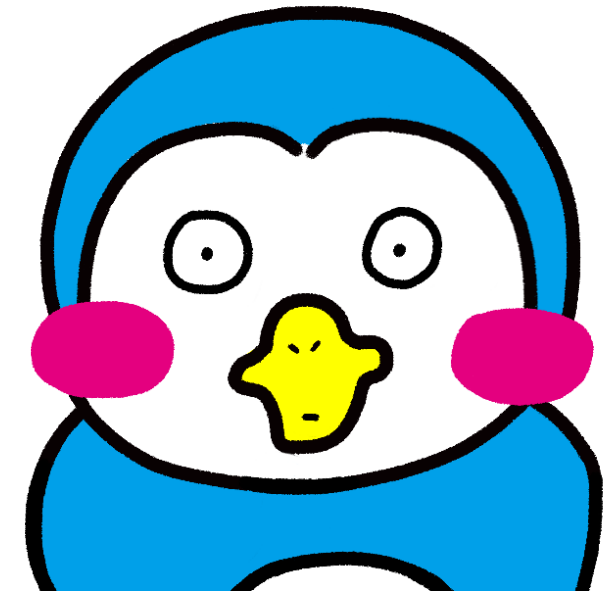


콘솔(console)은 브라우저의 디버깅 콘솔을 의미한다. 콘솔은 브라우저 안에 내장된 브라우저의 하위 객체이므로 브라우저 객체를 통해 접근할 수 있다.

객체 뒤에 . 을 찍으면  
객체가 가지고 있는 데이터에  
접근할 수 있다

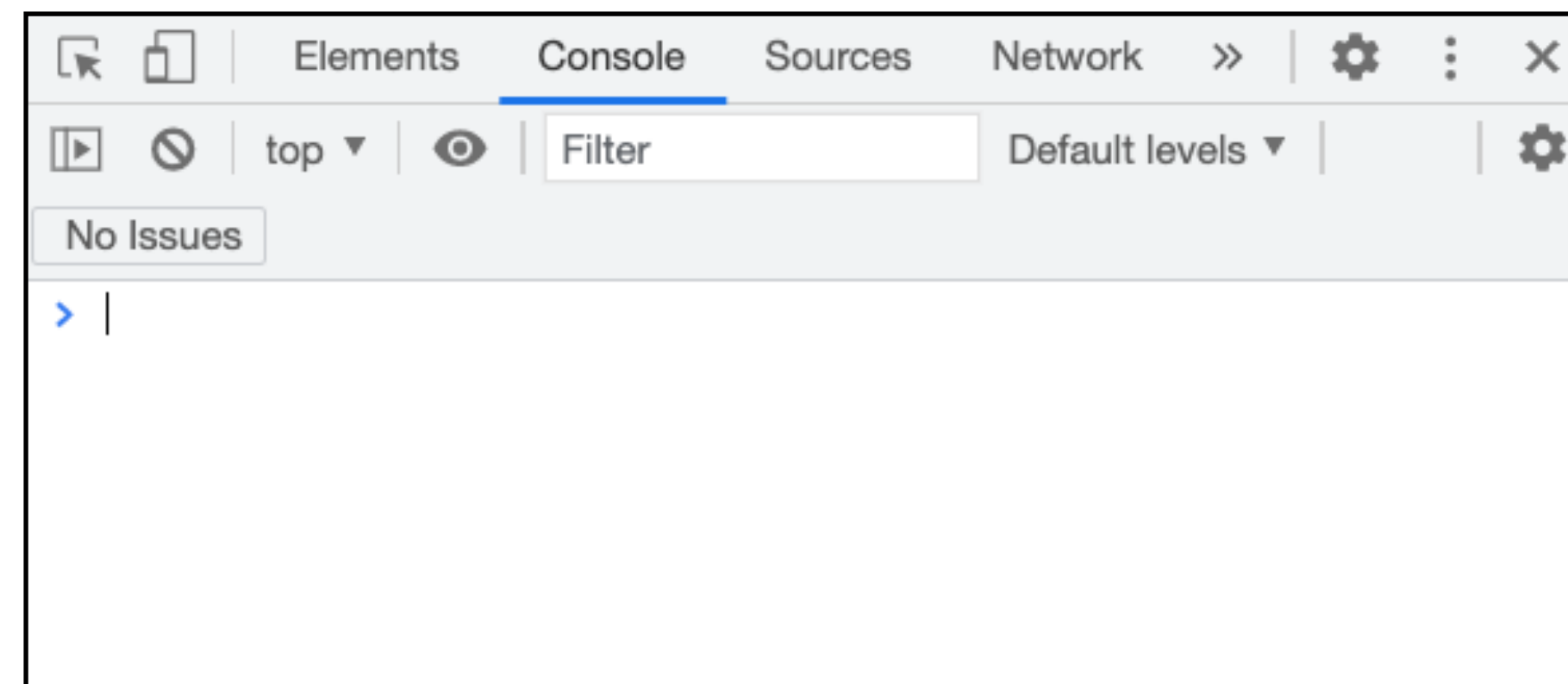
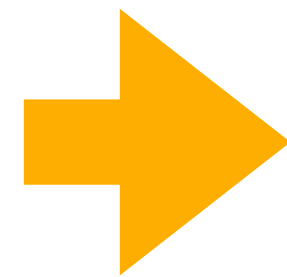
# window.console

# 디버깅 콘솔이 뭔데



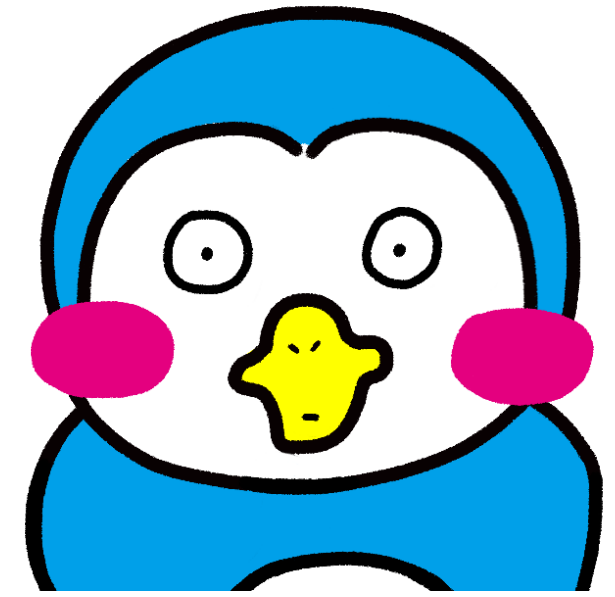
웹브라우저의 '개발자 도구'를 열면 메뉴 중 '콘솔(console)'이라는 항목이 존재하는데, 이것을 누르면 나타나는 화면이 바로 디버깅 콘솔이다.

```
...<!DOCTYPE html> == $0
<html lang="en-US" prefix="og: https://ogp.me/ns#">
  ><head>...</head>
  ><body>
    ><div id="root">...</div>
    ><script type="application/json" id="hydration">...</script>
  </body>
</html>
```



=> 디버깅 콘솔을 이용하면 자바스크립트 코드를 테스트(점검)할 수 있다!

# 충격적인(?) 사실이지만 꿀팁



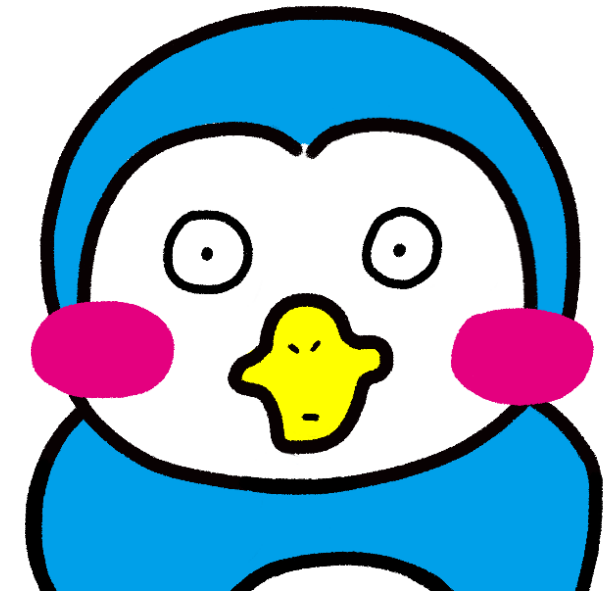
우리가 웹브라우저에 접근할 수 있게 해주는 객체 window는 console을 비롯한 다양한 하위 객체 및 데이터, 메소드 등을 포함하고 있는데, 사용자는 편의를 위해 window. 을 생략한 채로 코드를 작성할 수 있다.

window.console ← 이렇게 쓰나

console ← 이렇게 쓰나 똑같음

# 자바스크립트 코드 작성 규칙

---

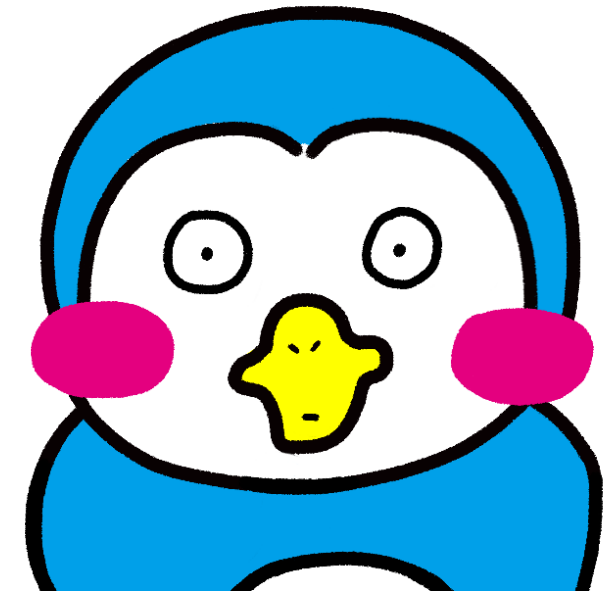


왠지 이쯤부터 본격적인 코드 작성이 시작된다는 느낌이 들어서...

- 대문자와 소문자를 잘 구분하여 작성한다.
- 구문의 끝에는 세미콜론을 입력해 '구문의 끝'을 알릴 수 있다. 이는 선택 사항이다.
- 가능하면 한 줄에는 두 개 이상의 구문을 쓰지 않는다.

# 내용 정리

---



- console 객체는 브라우저의 디버깅 콘솔에 접근할 수 있게 해 준다.
- console 객체는 window의 하위 객체이다.
- 자바스크립트 코드에서, window. 은 생략이 가능하다.
- console.log(x) 는 'x를 콘솔에 기록해두자'라는 기능의 메소드 호출이다.
- 코드에서 세미콜론(;)은 '구문의 끝'을 의미하는 기호이다. 세미콜론의 사용은 선택사항이다.