

7. 생성자 함수

(1) **prototype**: 생성자 함수로 생성된 객체가 공통으로 가지는 공간

ex1 - 학생들의 이름과 국어점수, 영어점수, 총점, 평균 출력하기

```
<script>
function Student(name, kor, eng) { // 생성자 함수 (이름, 국어점수, 영어점수)
    this.name = name;
    this.kor = kor;
    this.eng = eng;
}
// prototype (총점, 평균)
Student.prototype.getSum = function () { return this.kor + this.eng + this.mat; };
Student.prototype.getAvg = function () { return this.getSum() / 3 * 10 / 10; };
```

(2) **캡슐화**: 특정 속성이나 메소드를 사용자가 바로 사용할 수 없게 막는 기술
(숨겨놓은 속성은 getter와 setter를 이용하여 간접 접근 가능)

ex2 - 직사각형의 넓이 출력

캡슐화 전	캡슐화 후 (width와 height 숨기기)
<pre><script> function Rectanle(width, height) { this.width = width; this.height = height; } Rectanle.prototype.getArea = function () { return this.width * this.height; }; var rectangle = new Rectanle(5, 7); alert('AREA : ' + rectangle.getArea()); </script></pre>	<pre><script> function Rectangle(w, h) { var width = w; var height = h; // 숨긴 width와 height를 // getter를 이용하여 가져온다. this.getWidth = function () { return width; }; this.getHeight = function () { return height; }; }</pre>

(3) **상속**: 기존 생성자 함수나 객체를 기반으로 새로운 것을 만들 때 더 쉽게 만드는 것

ex3 - ex2의 Rectangle 함수에서 상속받은 Square 함수

```
function Square(length) {
    this.base = Rectangle;
    this.base(length, length);
}
Square.prototype = Rectangle.prototype;
```