untitled

```
/**
 * Class => Weapon(name, damage)
 * -----------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @param {number} damage   The weapon's damage.
 * @property {number} damage
 */



/**
 * Weapon Extends Item Class
 * -------------------------
 */
```

untitled

```
1  /**
2   * Class => Weapon(name, damage)
3   * ----------------------------------
4   * Creates a weapon item.
5   * Weapon items can be equipped for use in battle.
6   *
7   * The Weapon class constructor will call
8   *   the super class (Item) constructor
9   *   while passing in the 1 Item constructor param
10  *
11  * @name Weapon
12  * @param {string} name      The weapon's name.
13  * @param {number} damage    The weapon's damage.
14  * @property {number} damage
15  */
16
17 function Weapon () {
18
19 }
20
21 /**
22  * Weapon Extends Item Class
23  * ------------------------------------
24  */
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

Class Name
@name

```
/**
 * Class => Weapon(name, damage)
 * ------------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @param {number} damage   The weapon's damage.
 * @property {number} damage
 */

function Weapon (name, damage) {

}

/**
 * Weapon Extends Item Class
 * --------------------------
 */
```

untitled

```
/**
 * Class => Weapon(name, damage)
 * ------------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @param {number} damage   The weapon's damage.
 * @property {number} damage
 */

function Weapon (name, damage) {

  this.damage = damage;

}

/**
 * Weapon Extends Item Class
 * -------------------------
 */
```

Parameters
@param

Property
@property

Property Assignment

untitled

```
/**
 * Class => Weapon(name, damage)
 * ------------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @p                       The weapon's damage.
 * @property {number} damage
 */

function Weapon (name, damage) {

  this.damage = damage;

  Item.call(this, name);
}

/**
 * Weapon Extends Item Class
 * --------------------------
 */
```

Super Class => Item

Super class constructor Parameter
@param

Context => Weapon Class

invoke super class constructor method

untitled

```
/**
 * Class => Weapon(name, damage)
 * ------------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @param {number} damage   The weapon's damage.
 * @property {number} damage
 */

function Weapon (name, damage) {

  this.damage = damage;

  Item.call(this, name);
}

/**
 * Weapon Extends Item Class
 * ------------------------------
 */

Weapon.prototype = Object.create(Item.prototype);
```

Sub Class

Super Class

Sub Class extends Super Class prototype
Sub Class inherits all Super Class methods

untitled

```
/**
 * Class => Weapon(name, damage)
 * ------------------------------
 * Creates a weapon item.
 * Weapon items can be equipped for use in battle.
 *
 * The Weapon class constructor will call
 *   the super class (Item) constructor
 *   while passing in the 1 Item constructor param
 *
 * @name Weapon
 * @param {string} name     The weapon's name.
 * @param {number} damage   The weapon's damage.
 * @property {number} damage
 */

function Weapon (name, damage) {

  this.damage = damage;

  Item.call(this, name);
}

/**
 * Weapon Extends Item Class
 * -------------------------
 */

Weapon.prototype = Object.create(Item.prototype, {
  constructor: {
    value: Item
  }
});
```

Inherits the Super Class
constructor method