بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

تست کردن کتابخانه ی prototype

مدرس : مهندس افشین رفوآ

تست کردن کتابخانه ی prototype

ضمیمه کردن کتابخانه ی Prototype به صفحه ی وب

برای استفاده از یک کتابخانه یا framework جاوا اسکریپت، باید آنرا به صفحه ی وب خود اضافه نمایید.

به منظور افزودن کتابخانه به صفحه ی وب، باید از تگ <script> استفاده کرده و خاصیت src آن را به URL کتابخانه متصل (set) کنید:

```
<!DOCTYPE html>
<html>
<head>
<script src="prototype.js"></script>
</head>

<body>
```

توضحیاتی درباره ی کتابخانه ی Prototype

کتابخانه Prototype توابعی را برای برنامه نویسی سهل، عرضه می کند.

همانند jQuery، کتابخانه ی Prototype نیز تابع $() را در خود دارد.

تابع $()معادل متد document.getElementById() در جاوا اسکریپت بوده و کارکردی مشابه آن دارد. اگر با توجه به مدل DOM در HTML، شناسه یک عنصر به آن فرستاده شود، در کتابخانه ی Prototype به همه ی خواص و توابع آن دسترسی خواهید داشت.

برخلاف jQuery، کتابخانه Prototype تابع ready() را ندارد. بجای آن در Prototype افزونه هایی (extension) به مرورگر و مدل DOM اضافه گردیده است.

تخصیص یک تابع به رخداد onload:

در جاوا اسکریپت، می توان یک تابع را به یک رخدادی (event) مانند onload انتساب (تخصیص) داد:

روش جاوا اسکریپت:

```
function myFunction() {
  var obj = document.getElementById("h01");
  obj.innerHTML = "Hello Prototype";
}
onload = myFunction;
```

معادل prototype آن:

```
function myFunction() {
  $("h01").insert("Hello Prototype!");
}
Event.observe(window, "load", myFunction);
```

Event.observe() سه آرگومان می گیرد:

آرگومان اول (window): شی DOM یا BOM ای است که می خواهید مدیریت کنید (روی آن عملیاتی انجام دهید).

آرگومان دوم (load): رخدادی است که می خواهید اداره کنید (با اجرای آن کاری انجام شود).

آرگومان سوم (myFunction): تابعی است که می خواهید فراخوانی شود.

کار با کتابخانه ی prototype

مثال زیر را امتحان کنید:

```
<!DOCTYPE html>
```

```
<html>
<head>
  <title></title>
<script src="prototype.js"></script>
  <script>
    function myFunction() {
      $("h01").insert("Hello Prototype!");
    }
    Event.observe(window, "load", myFunction);
  </script>
</head>
<body>
  <h1 id="h01"></h1>
</body>
</html>
```

حال این مثال را امتحان کنید:

```
<!DOCTYPE html>
<html>
<head>
  <title></title>
<script src="prototype.js"></script>
  <script>
    function myFunction() {
      $("h01").writeAttribute("style", "color:red").insert("Hello Prototype!");
    }
    Event.observe(window, "load", myFunction);
  </script>
</head>
<body>
  <h1 id="h01"></h1>
</body>
</html>
```

همان طور که در مثال بالا مشاهده می کنید، prototype همانند jQuery به شما اجازه ی انجام چندین کار به طور همزمان روی یک شی را می دهد (امکان chaining را فراهم می نماید).