

# JQuery

**JQuery** merupakan suatu **framework** (library) Javascript yang menekankan bagaimana interaksi antara **Javascript** dan **HTML**. JQuery pertama kali dirilis pada tahun 2006 oleh John Resig. Pada perkembangannya JQuery tidak sekedar sebagai framework Javascript, namun memiliki kehandalan dan kelebihan yang cukup banyak. Hal tersebut menyebabkan banyak developer web menggunakannya. JQuery memiliki slogan “**Write less, do more**” yang kurang lebih maksudnya adalah kesederhanaan dalam penulisan code, tapi dengan hasil yang lebih banyak.

JQuery merupakan library open source dengan lisensi GNU General Public License dan MIT License. Dari sisi ukurannya, framework JQuery sungguh ramping, hanya sekitar 20 KB dan hanya terdiri dari satu file. Namun demikian, bagi yang menginginkan fungsi lebih, JQuery memungkinkan penambahan fungsionalitas dalam bentuk plugin. Saat ini tersedia ribuan plugin yang dapat diperoleh secara gratis di internet.

## Apa yang bisa dilakukan dengan JQuery?

### 1. Mengakses bagian halaman tertentu dengan mudah.

Tanpa adanya library Javascript khusus, untuk mengakses suatu bagian tertentu dari halaman, harus mengikuti aturan **Document Object Model (DOM)** dan pengaksesan harus secara spesifik menyesuaikan dengan struktur HTML. Dengan kata lain, pengaksesan bagian tertentu dari halaman sangat tergantung pada struktur dari HTML. JQuery menawarkan cara yang mudah (bahkan sangat mudah) dalam mengakses bagian tertentu dari halaman. Pengaksesan juga tidak terlalu bergantung pada struktur HTML.

### 2. Mengubah tampilan bagian halaman tertentu.

CSS (Cascading Style Sheet) menawarkan metode yang cukup handal dalam mengatur dan mempercantik halaman web. Namun terkadang CSS punya kelemahan yang cukup mengganggu, yaitu beberapa perintah CSS tidak didukung oleh semua browser. Cukup merepotkan jika kita harus mendesign halaman web dengan beberapa CSS sekaligus. Sekali lagi JQuery menawarkan solusi untuk mengatasi hal tersebut. Dengan JQuery, “kesenjangan” yang terjadi antara browser dalam urusan CSS akan tertutup dengan baik.

### 3. Mengubah isi dari halaman.

Jaman dulu (baca:sebelum JQuery lahir) cukup sulit jika kita akan mengubah sebagian isi dari halaman. Mengubah disini dapat berarti mengganti teks, menambahkan teks atau gambar, mengurutkan suatu daftar (list), menghapus baris tabel dan sebagainya. Dengan JQuery, hal tersebut dapat dilakukan dengan hanya beberapa baris perintah.

#### 4. Merespond interaksi user dalam halaman.

Website yang baik tidak cukup digambarkan dengan user-interface dan tampilan yang memukau. Namun lebih dari itu, bagaimana pengunjung dapat berinteraksi dengan website dan dapat mengatur tampilannya sendiri. Interaktivitas sangat bergantung bagaimana pemrograman yang dipakai dalam menangani event-handling. Javascript sendiri memiliki beberapa event-handling seperti onclick untuk menangani event saat terjadi click. Namun demikian, event handling pada Javascript terbatas pada object-object tertentu, dan jenisnya pun terbatas. JQuery melengkapi semuanya dengan tambahan penanganan event-handling yang semakin mudah.

#### 5. Menambahkan animasi ke halaman.

Animasi seringkali disertakan dalam suatu halaman web untuk menambah kecantikannya. Saat ini animasi masih cukup digemari oleh para peselancar situs. Animasi dapat dibuat dalam berbagai gaya, ada yang menggunakan Flash, gambar bergerak (GIF), video, dan sebagainya. Masing-masing tentu memiliki kelebihan dan kekurangan masing-masing. JQuery sendiri menawarkan konsep animasi (walaupun masih sederhana) yang cukup apik namun ramah bandwidth alias ringan. Salah satu animasi yang bisa dibuat dengan JQuery adalah fading jika terdapat suatu bagian dari halaman ditambahkan atau dihilangkan.

#### 6. Mengambil informasi dari server tanpa me-refresh seluruh halaman.

Mengambil informasi dari server tanpa refresh halaman merupakan salah satu konsep dasar dari yang namanya **AJAX (Asynchronous Javascript and XML)**. Pada penerapannya, cukup ribet jika harus membangun website dengan konsep AJAX, saat ini banyak library khusus yang berusaha mempermudahnya. JQuery merupakan salah satunya.

#### 7. Menyederhanakan penulisan Javascript biasa.

Semboyan JQuery adalah “**Write less, do more**” atau dengan kata lain kesederhanaan dalam penulisan code, tetapi menghasilkan tampilan yang lebih. Sebenarnya inilah yang menjadi daya tarik tersendiri buat para pengembang web untuk menggunakan JQuery.

Contoh :

```
<!DOCTYPE html>
<html>
<head>
  <script src="jquery-1.9.1.js"></script>
  <script>
    $(document).ready(function(){
      $("p").click(function(){
        $(this).hide();
      });
    });
  </script>
</head>
<body>
  <p>Klik saya, dan saya akan hilang</p>
  <p>Klik saya juga</p>
```

```
<p>Klik saya juga</p>
</body>
</html>
```

jQuery Selectors.

Selector	Example	Selects
*	\$("*")	All elements
#id	\$("#lastname")	The element with id="lastname"
.class	\$(".intro")	All elements with class="intro"
.class,.class	\$(".intro,.demo")	All elements with the class "intro" or "demo"
element	\$("p")	All <p> elements
el1,el2,el3	\$("h1,div,p")	All <h1>, <div> and <p> elements
:first	\$("p:first")	The first <p> element
:last	\$("p:last")	The last <p> element
:even	\$("tr:even")	All even <tr> elements
:odd	\$("tr:odd")	All odd <tr> elements
:first-child	\$("p:first-child")	All <p> elements that are the first child of their parent
:first-of-type	\$("p:first-of-type")	All <p> elements that are the first <p> element of their parent
:last-child	\$("p:last-child")	All <p> elements that are the last child of their parent
:last-of-type	\$("p:last-of-type")	All <p> elements that are the last <p> element of their parent
:nth-child(n)	\$("p:nth-child(2)")	All <p> elements that are the 2nd child of their parent
:nth-last-child(n)	\$("p:nth-last-child(2)")	All <p> elements that are the 2nd child of their parent, counting from the last child
:nth-of-type(n)	\$("p:nth-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent
:nth-last-of-type(n)	\$("p:nth-last-of-type(2)")	All <p> elements that are the 2nd <p> element of their parent, counting from the last child
:only-child	\$("p:only-child")	All <p> elements that are the only child of their parent

:only-of-type	\$("p:only-of-type")	All <p> elements that are the only child, of its type, of their parent
parent > child	\$("div > p")	All <p> elements that are a direct child of a <div> element
parent descendant	\$("div p")	All <p> elements that are descendants of a <div> element
element + next	\$("div + p")	The <p> element that are next to each <div> elements
element ~ siblings	\$("div ~ p")	All <p> elements that are siblings of a <div> element
:eq( <i>index</i> )	\$("ul li:eq(3)")	The fourth element in a list (index starts at 0)
:gt( <i>no</i> )	\$("ul li:gt(3)")	List elements with an index greater than 3
:lt( <i>no</i> )	\$("ul li:lt(3)")	List elements with an index less than 3
:not( <i>selector</i> )	\$("input:not(:empty)")	All input elements that are not empty
:header	\$(":header")	All header elements <h1>, <h2> ...
:animated	\$(":animated")	All animated elements
:focus	\$(":focus")	The element that currently has focus
:contains( <i>text</i> )	\$(":contains('Hello'))"	All elements which contains the text "Hello"
:has( <i>selector</i> )	\$("div:has(p)")	All <div> elements that have a <p> element
:empty	\$(":empty")	All elements that are empty
:parent	\$(":parent")	All elements that are a parent of another element
:hidden	\$("p:hidden")	All hidden <p> elements
:visible	\$("table:visible")	All visible tables
:root	\$(":root")	The document's root element
:lang( <i>language</i> )	\$("p:lang(de)")	All <p> elements with a lang attribute value starting with "de"
[ <i>attribute</i> ]	\$("[href]")	All elements with a href attribute
[ <i>attribute=</i> <i>value</i> ]	\$("[href='default.htm']")	All elements with a href attribute value equal to "default.htm"
[ <i>attribute!=</i> <i>value</i> ]	\$("[href!='default.htm']")	All elements with a href attribute value not equal to "default.htm"

[attribute\$=value]	\$("[href\$='.jpg']")	All elements with a href attribute value ending with ".jpg"
[attribute =value]	\$("[title = 'Tomorrow']")	All elements with a title attribute value equal to 'Tomorrow', or starting with 'Tomorrow' followed by a hyphen
[attribute^=value]	\$("[title^='Tom']")	All elements with a title attribute value starting with "Tom"
[attribute~=value]	\$("[title~='hello']")	All elements with a title attribute value containing the specific word "hello"
[attribute*=value]	\$("[title*='hello']")	All elements with a title attribute value containing the word "hello"
:input	\$(":input")	All input elements
:text	\$(":text")	All input elements with type="text"
:password	\$(":password")	All input elements with type="password"
:radio	\$(":radio")	All input elements with type="radio"
:checkbox	\$(":checkbox")	All input elements with type="checkbox"
:submit	\$(":submit")	All input elements with type="submit"
:reset	\$(":reset")	All input elements with type="reset"
:button	\$(":button")	All input elements with type="button"
:image	\$(":image")	All input elements with type="image"
:file	\$(":file")	All input elements with type="file"
:enabled	\$(":enabled")	All enabled input elements
:disabled	\$(":disabled")	All disabled input elements
:selected	\$(":selected")	All selected input elements
:checked	\$(":checked")	All checked input elements