

سئوتکنیکال: بخش html basic & Other

بخش اول: HTML Basic

برای اینکه وب سایت ما از دید کراولر و موتور جست و جو crawlable باشد، باید شرایط html ای زیر را داشته باشد.

● <title>

مهم ترین تگ سئویی موجود است و باید در head وب سایت قرار گیرد. هم چنین باید دسترسی تغییر آن در پنل ادمین داده شود. در صورت خالی ماندن این تگ، باید به طور پیش فرض از روی عنوان (h1) برداشته شود.

● "meta name="description"

بعد از تگ title، مهم ترین تگ سئویی موجود در بخش head وب سایت است. علاوه بر قرارگیری در قسمت head باید دسترسی تغییر در پنل ادمین داده شود. در صورت خالی ماندن این تگ، باید به طور پیش فرض از 160 کاراکتر ابتدایی متن برداشته شود.

● "meta name="robots"

متاتگ robots باید در قسمت <head> وب سایت حضور داشته باشد. این متاتگ در حال بیس دارای ویژگی index, follow است که باید امکان دسترسی برای تبدیل هر کدام به noindex و nofollow در پنل ادمین وجود داشته باشد.

به صورت پیش فرض "Meta name="robots" content="index,follow,max-image-preview:large"

● <h1> <h2> ,,,, <h6>

برای درک بهتر گوگل از صفحه بندی وب سایت، ما از عنوان و زیر عنوان های زیر استفاده می کنیم. باید در ادیتور موجود در پنل، امکان تبدیل جمله و عبارت به h2 تا h6 وجود داشته باشد.

باید به این نکته دقت کنید که هر صفحه در بهترین حالت ممکن فقط دارای یک h1 می باشد. همچنین در فوتر برای عنوان ها از h3 استفاده شود.

●

برای انتقال اعتبار صحیح و درک صحیح موتور جست و جو از لینک، باید به جای استفاده از جاوا اسکریپت، از تگ a استفاده کرد. بنابراین باید این مورد در لینک سازی داخلی رعایت شود. البته در لینک هایی که از دید سئویی اهمیتی ندارند و همزمان با تعامل کاربر تغییر می کند (همانند فیلترهای سایت های فروشگاه و هنگام پروموت محصول و یا استفاده از url parameter) می توان از جاوا اسکریپت استفاده نمود.

• Alt attribute

هر عکس برای درک بهتر توسط افراد نابینا دارای یک ویژگی متنی به اسم متن جایگزین می باشد. تمام عکس های موجود در وب سایت باید دارای این ویژگی باشد و موتور جست و جو نیز برای این متن جایگزین اهمیت بالایی قائل است.

```


با قرار گرفتن متا تگ زیر در قسمت head وب سایت، در شبکه های اجتماعی توییتر، لینک صفحه به feature snippet تبدیل می شود. (اختیاری)

```
<meta name="twitter:card" content="summary_large_image" />
```

```
<meta name="twitter:label1" content="زمان تقریبی برای خواندن">
```

```
<meta name="twitter:data1" content="10 دقیقه">
```

## • "meta property="og:title">

با قرار گرفتن متا تگ زیر در قسمت head وب سایت، در شبکه های اجتماعی، لینک صفحه به feature snippet تبدیل می شود. (اختیاری)

```
<meta property="og:title" content="تایتل اصلی صفحه باید باشد" />
```

```
<meta property="og:description" content="متا دیسکریپشن صفحه باید باشد">
```

## • </""=link rel="canonical"href>

با تگ بالا، ما به موتور جست و جو نشان می دهیم که محتوای اصلی در آن موضوع در کدام صفحه قرار گیرد و از محتوای تکراری جلوگیری می کنیم. این تگ باید در قسمت head وب سایت قرار گرفته و دسترسی تغییر آن در پنل داده شود. در حالت پیش فرض نیز کنونیکال روی خود صفحه مد نظر می باشد.

```
<link rel="canonical" href="example.com" />
```

## • "meta name="viewport">

متا تگ زیر نیز برای ایجاد حالت ریسپانسیو باید در قسمت head وب سایت قرار گیرد.

```
<meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
```

## بخش دوم: Semantic Tags

سمانتیک تگ ها در نسخه html5 معرفی گردیدند و در درک درست موتور جست و جو درباره صفحه بندی و معنای هر کدام تاثیرگذاری فراوانی دارد. (همه این موارد در قسمت body حضور دارند.)

**<article>**

این تگ باید در قسمت های محتوایی متنی وب سایت همچون بلاگ و اخبار و ... مورد استفاده قرار گیرد. قسمت اصلی محتوای هر مقاله، باید در این تگ قرار گیرد.

**<aside>**

این تگ باید در هنگام وجود سایتد بار در هر قسمت از وب سایت وجود داشته باشد.

**<figure>**

برای جدا کردن محتوای تصویری همانند عکس از محتوای دیگر مورد استفاده قرار می گیرد.  
برای مثال هر عکس از وب سایت در حالت زیر باشد:

**<figure>**

****

**<figcaption>**      **<figcaption>**

**</figure>**

**<figcaption>**

برای قرار دادن کپشن روی محتوای تصویری استفاده می گردد و به نوعی زیر شاخه figure می باشد. (اختیاری)

**<footer>**

فوتر وب سایت باید به جای استفاده از div دارای این تگ باشد.

**<header>**

هدر وب سایت باید به جای استفاده از div دارای این تگ باشد.

**<main>**

بدنه اصلی محتوای هر صفحه (به غیر از header و footer) باید در این قسمت قرار گیرد.

## <nav>

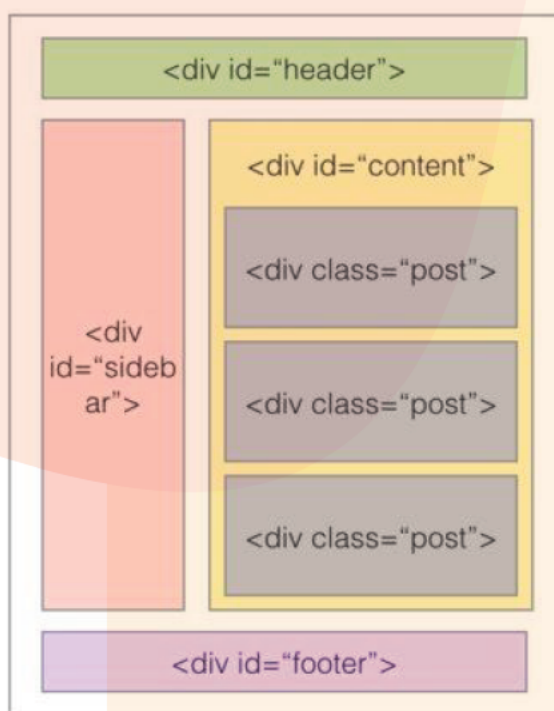
هنگامی که منو لینک بیلدینگ در هر قسمت از وب سایت قرار گیرد، باید در تگ `nav` قرار گیرد. برای مثال در قسمت `header` وب سایت، ما دارای یک منو هستیم که باید در تگ `<nav>` قرار گیرد.

## <section>

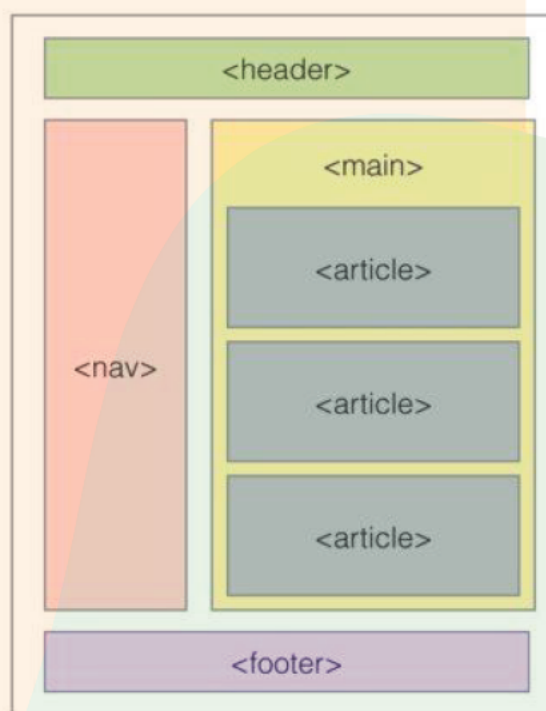
هر بخش در وب سایت که معنا و مفهوم متفاوتی نسبت به بخش دیگر دارد، باید در این تگ قرار بگیرد.

## HTML4 vs HTML5 Page Structure on a Blog

### HTML4: Lots of Classes/IDs



### HTML5: Semantic Tags/Sections





## بخش سوم: Sitemap

یک سایت مپ مناسب و کامل باید امان های زیر را داشته باشد و به صورت داینامیک باشد. (بعد از قرار گرفتن هر محتوا در وب سایت، در سایت مپ مربوطه قرار گیرد.)

### • <sitemapindex>

اگر وب سایت دارای چندین سایت مپ متنوع باشد، باید در تگ بالا قرار گیرد.

### • <sitemap>

هر کدام از سایت مپ ها در تگ زیر قرار گیرد.

برای مثال در حال حاضر دو سایت مپ زیر در sitemapindex قرار گرفته است.

```
<sitemapindex>
<sitemap>
<loc>https://websima.academy/sitemap-misc.xml</loc>
<lastmod>2021-07-28T06:02:44+00:00</lastmod>
</sitemap>
```

**<sitemap>**

**<loc>https://websima.academy/sitemap-tax-glossarys.xml</loc>**

**<lastmod>2021-07-28T06:02:44+00:00</lastmod>**

**</sitemap>**

**<urlset>**

هر سایت مپ xml با تگ زیر شروع می شود.

**<url> •**

هنگامی که بخواهیم یک صفحه از وب سایت را در سایت مپ نشان دهیم، از تگ زیر استفاده می کنیم.

**<loc> •**

برای نشان دادن url صفحه از این تگ استفاده می کنیم.

**<lastmod> •**

نشان دهنده آخرین تغییرات آن صفحه می باشد که می تواند از طریق تاریخ آخرین تغییر ایجاد شده در پنل مشخص شود.

**<changefreq> •**

نشان دهنده بازه تغییر محتوای آن صفحه می باشد.

**<priority> •**

میزان اهمیت صفحه می باشد که عددی بین 0 تا 1 می باشد.

**برای مثال، در حال حاضر در sitemap-misc وب سایت وب سیما، المان زیر برای صفحه اصلی قرار دارد.**

**<urlset**

**<url>**

**<loc>https://websima.academy/</loc>**

<lastmod>2021-07-10T06:54:54+00:00</lastmod>

<changefreq>daily</changefreq>

<priority>1.0</priority>

</url>

<urlset>

## بخش چهارم: Rendering Method

رندرینگ وب سایت به 3 صورت زیر صورت می گیرد.

Server side rendering

Hybrid rendering

Dynamic rendering

در بهترین حالت ممکن در صورت استفاده از لایبرری و فریم ورک های جاوا اسکریپت، باید وب سایت در حالت server side rendering باشد، ولی در صورت نبود شرایط، در حداقل ترین شرایط ممکن وب سایت باید دارای hybrid rendering باشد تا همان های مهم سنوپی crawlable باشد.

## بخش پنجم: Crawl Budget Optimization

url parameter

از این ویژگی بیشتر در سایت های فروشگاه های و با هدف ایجاد فیلتر استفاده می گردد. Url parameter باعث ایجاد محتوای تکراری می گردد و می تواند به بودجه خزش وب سایت آسیب وارد کند. برای جل این موضوع باید دسترسی این موارد در بخش url parameter سرچ کنسول بسته شود و به صورت زیر url پارامتر بندی شود.

### 1. Eliminate Unnecessary Parameters

از قرار دادن پارامترهای اضافی در url خودداری کنید.



https://www.example.com/page?sessionID=123456



https://www.example.com/page

## 2. Prevent Empty Values

تا زمان کال نشدن پارامترها و نبودن value از نمایش parameter های اضافی در url جلوگیری کنید.

❌ <https://www.example.com/page?key1=value1&key2=&key3=>

✅ <https://www.example.com/page?key1=value1>

## 3. Use Keys Only Once

در هنگام انتخاب چند value از یک key, آدرس زیر باید به صورت زیر ساخته شود.

❌ <https://www.example.com/page?key1=value1&key1=value2>

✅ <https://www.example.com/page?key1=value1-value2>

## 4. Order URL Parameters

Key ها باید به صورت سورت بندی شده در url قرار گیرد تا از duplicate content جلوگیری شود.

❌ <https://www.example.com/page?key2=value2&key1=value1>

✅ <https://www.example.com/page?key1=value1&key2=value2>

## بخش ششم: others

### قسمت اول: Breadcrumb

در تمامی صفحات وب سایت، باید از بردهکرامب و ساختار درختی آن استفاده گردد.

### قسمت دوم: mobile friendly

باید وب سایت در نسخه موبایل به صورت ریسپانسیو باشد و تمام المان های اصلی صفحات در نسخه دسکتاپ در نسخه موبایل هم حضور داشته باشد. علاوه بر موارد بالا از نزدیک بودن المان های لینک، کوچک بودن سایز فونت و فیت بودن سایز صفحه در نسخه موبایل نیز اطمینان حاصل شود.

### قسمت سوم: url structure

باید در پنل کاربری و ادمین، دسترسی تغییر بعد از / داده شود. برای مثال به فرض قرار گرفتن یک محصول بتوان این قسمت از url را تغییر داد.

`example.com/shop/product/*****`

### قسمت چهارم: hsts

با فعال سازی این قابلیت در کنترل پنل هاست یا سرور، به شخص اجازه داده نمی شود که در صورت وجود نسخه http از وب سایت دیدن شود و به امنیت کاربران کمک می کند. در صورت استفاده از کنترل پنل cpanel با قرار دادن کد زیر در htaccess این مورد حل خواهد شد.

```
Security header Enable HSTS
```

```
Header always set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
```

```
env=HTTPS
```

### قسمت هفتم: http to https

پس از مشخص شدن آنکه وب سایت بدون www و یا با www قرار است که به کاربران نمایش داده شود. اقدامات زیر باید انجام گیرد. (پیشنهاد ما استفاده از نسخه بدون www می باشد.)

<https://www.example.com> redirect 301 to <https://example.com>

<http://www.example.com> redirect 301 to <https://example.com>

<http://example.com> redirect 301 to <https://example.com>

### قسمت هشتم: redirection

در پنل ادمین، باید بستری برای ریدایرکت در انواع مختلف 301,302,307 وجود داشته باشد.

## سئوتکنیکال: بخش page experience

درصد تاثیرگذاری	عدد پیشنهادی گوگل	تغریف	معیار
15%	0.9s	زمان لود شدن اولین المان صفحه	FCP
25%	1.2s	زمان لود شدن بزرگترین المان صفحه	LCP
25%	150ms	میزان زمان بلاک شدن رندر	TBT
15%	1.3s	زمان لود شدن المان های بالای صفحه (شامل هدر و اسلایدر)	SI
15%	2.5s	زمان مورد نیاز برای ایجاد شرایط تعامل کاربر	TTI
5%	0.1	حرکت المان های صفحه مانند انیمیشن ها، عکس های بدون طول و عرض	CLS

برای تست می توان از gt metrix و یا page speed insight استفاده نمود.

لیست زیر مجموعه اقداماتی هستند که در صورت انجام آن توسط تیم دولوپ و تیم کانفیگ سرور به بالاترین حد بهینه سازی خواهیم رسید.

### بخش اول: server

#### قسمت اول: TTFB

- ریدایرکت 301 های از نسخه http به https حتماً توسط سرور انجام شود.
- یک سرور با کیفیت به همراه ping و uptime بالا الزامی است.

- حتی از سرور نزدیک به محل کاربران استفاده گردد. (در صورت عدم استفاده از CDN)
- در موارد پیشرفته، اقدامات زیر در بک اند وب سایت الزامی است.

#### Normalizing a Database

#### Database Indexes

#### Denormalization of a Database

#### قسمت دوم: Caching

- Browser caching

حتما این قابلیت در وب سرور اکتیو شود و کچینگ ما از نوع cache control header باشد.

- Server caching

در حالت ساده می توان این قابلیت را به وسیله فعالسازی cdn فعال نمود.

#### قسمت سوم: CDN

حتما از cdn در وب سایت استفاده گردد. می توان از cdn کلودفلر نسخه رایگان نیز برای سرورهای خارج از ایران استفاده نمود. (برای سرورهای ایرانی مشکل زاست).

#### قسمت چهارم: HTTP2 Response

#### قسمت پنجم: Keep-Alive

#### قسمت ششم: GZIP compression

البته در کنار gzip می توان از br نیز برای فشرده سازی فایل ها استفاده نمود.

#### بخش دوم: HTML

تمامی فایل های html وب سایت باید minify گردد.

#### بخش سوم: CSS

#### قسمت اول: Avoid RBR on CSS

برای حل این مشکل اقدامات زیر لازم است:

باز کردن بخش Inspect مرورگر و اضافه کردن تولز coverage

پیدا کردن فایل های CSS موجود اعلام شده و بررسی کدهای استفاده شده و نشده در صفحه (یکی از فایل ها به صورت اکسترنال می باشد که باید به اینترنال تبدیل شود.)

مشخص شدن کد المان های اصلی صفحه (هدر، اسلایدر و فوتر در دو نسخه موبایل و دسکتاپ) و ذخیره آن به نام فایل

**critical.css در سرور**

سیو کردن بقیه کدهای موجود با نام **uncritical.css** در سرور

قرار دادن کد **critical.css** به صورت زیر در **<head>**

```
<link rel="preload" type="text/css" as="style" href="critical.css" rel="stylesheet" />
```

قرار دادن کد **uncritical.css** به صورت زیر در **<head>**

```
<link rel="preload" href="uncritical.css" as="style" onload="this.onload=null; this.rel='stylesheet'">
```

```
<noscript><link rel="stylesheet" href="uncritical.css"></noscript>
```

بهتر است که شخصی سازی حتی با قرار دادن ویژگی **media** ادامه دار باشد. به این گونه که اقدامات بالا در دو نسخه موبایل و دسکتاپ جدا (در توضیحات بالا با هم می باشد). انجام گیرد. برای مثال برای دو نسخه موبایل و دسکتاپ داشته باشیم:

```
<link rel="preload" type="text/css" as="style" href="mobilecritical.css" rel="stylesheet"
```

```
media="mobile"/>
```

```
<link rel="preload" type="text/css" as="style" href="desktopcritical.css" rel="stylesheet"
```

```
media="desktop"/>
```

**قسمت دوم: Avoid CSS @import**

**قسمت سوم: Minify CSS**

**قسمت چهارم: Reduce unused CSS**

**قسمت پنجم: Using CSS Sprites**

**Css sprite** تکنیکی است که برای کاهش درخواست ارسالی به سرور استفاده می شود. به وسیله آن میتوان عکس های کوچک وب سایت را ترکیب و به وسیله **css selector** کال شود.

**بخش چهارم: Java Script**

**قسمت اول: Avoid RBR on JS**

فایل های جاوا اسکریپت زیر باعث بلاک شدن رندر می گردد. در صورت امکان و ایجاد نکردن تجربه کاربری ضعیف (همانند لود نشدن اسلایدر، هدر و منو) به فایل های زیر ویژگی **async** یا **defer** اضافه گردد. (ترجیحا **defer**)

**قسمت دوم: Avoid document.write()**

**قسمت سوم: Minify JS**

**قسمت چهارم: Reduce unused JavaScript**

قسمت پنجم: Use ES6

قسمت ششم: Avoid Large JS Library

## بخش پنجم: Font

موارد زیر باید در لود کردن فونت های بالا رعایت گردد.

- تمام فونت ها باید در **سرور** ذخیره گردد.
- از حالت درختی (فایل اول باید لود گردد تا فایل دوم لود شود) خارج شود.
- تمام فونت ها به صورت `<link>` و با ویژگی `rel=preload` و همراه با `crosorigin="anonymous"` در قسمت `<head>` کال شود.
- اضافه کردن `display=swap` به انتهای فونت های خارجی (از جمله فونت گوگل)
- در صورت کال و انتخاب شدن فونت در فایل `css` به انتهای هر کد `font display=swap` اضافه گردد.

## بخش ششم: CLS

- تمام عکس های استفاده شده در سایت باید ویژگی **width** و **height** برایشان تنظیم شده باشد.
- از انیمیشن های سنگین و پر حجم استفاده نشود.

## بخش هفتم: Image

قسمت اول: Minify Img

- تمام عکس ها ابتدا باید به فرمت **webp** تبدیل گردند. سپس حجم آنها نیز به زیر 80k کاهش یابد.

قسمت دوم: Lazy Load Method

- تکنیک **lazy load** باید برای عکس های موجود در سایت مورد استفاده قرار گیرد. (یا به روش قرارگیری **فایل جاوا اسکریپت** و یا با قرار گرفتن ویژگی `loading=lazy` در `<img>`)

## بخش هشتم: Video

قسمت اول: Minify video

- در صورت استفاده از ویدئو در وب سایت، ابتدا در آپارات ذخیره و به صورت `embed` قرار گیرد و یا به صورت `webm` در وب سایت دربیاید.

## بخش نهم: Third Party

- مهم ترین منابع خارجی وب سایت، باید از طریق کدهای پایین در حالت `preload` قرار گیرد.
- اضافه کردن کد زیر به قسمت `<head>` (اگر هیچ کدام از فایل های پایین به صورت داخلی و در سرور لود نشود).

- <link rel="preconnect" href="example.com"
- <link rel="preconnect" href="example.com"
- <link rel="dnsprefetch" href="example.com"

## بخش دهم: DOM

برای جلوگیری از RBR، در امان های زیر باید رعایت گردد:

- Max Total DOM Elements=1500
- Maximum DOM Depth=32
- Maximum Child Elements=60

جمع امان های DOM نباید بیشتر از 1500 خط باشد.

جمع عمق امان های DOM نباید بیشتر از 32 امان باشد.

تعداد فرزندان هر امان نباید بیشتر از 60 باشد.



# پایان