

人物介绍Web页面

written by chaoqun

- HTML页面
- 购买按需计费云服务器
- 安装Web服务器
- 部署页面

总体来说，分为两步：一是完成纯html页面，二是部署页面可访问。

html

总体来说html是易于掌握的标记语言，但是如何正确的使用语义类的标签才是真正要去把握和理解的。

一般来说，可以把html标签分为3种类型：

- head中的元信息类标签
- img, video等替换类标签
- 还有语义类标签

很多语义类标签的视觉表现都差不多，section，nav和无语义的div基本相同。

但正确的使用语义类标签可以增强可读性，利于维护，也更适合被机器阅读，适合SEO。

- 自然语言表达能力的补充
- 文章标题和摘要
- 适合机器阅读和整体结构

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <meta http-equiv="X-UA-Compatible" content="ie=edge">
  <title>Personal Page</title>
</head>
<body>
  <header>
    <h1>关于我</h1>
    <cite>Nothing is true. Everything is possible. — Assassins
Creed</cite>
  </header>
```

```

<main>
  <article>
    <hgroup>
      <h2><em>Wu Chaoqun</em></h2>
      <h3>吴超群，中二，漫威，程序员</h3>
    </hgroup>
    <div>
      

      <p><strong>Thor & Rocky</strong>，海总 和 抖森，幻想举起喵喵锤</p>
    </div>
    <section>
      <h3>Here's a brief instruction:</h3>
      <p>本科<abbr title="Computer Science">CS</abbr>，研究生<abbr
title="Software Engineering">SE</abbr>，最开始是因为想制作游戏而选择计算机，后来也不知道搞什么去了。</p>
      <h3>书(推荐): </h3>
      <ul>
        <li>《深入理解计算机系统》</li>
        <li>《计算机程序的构造与解释》</li>
        <li>《编码：隐匿在计算机软硬件背后的语言》</li>
        <li>《妙趣横生的统计学》</li>
        <li>《鸟哥的 Linux 私房菜》</li>
        <li>《代码整洁之道》</li>
        <li>《编程珠玑》</li>
        <li>《数学之美》</li>
        <li>《Java 编程思想》</li>
        <li>《Effective Java》</li>
        <li>《黑客与画家》</li>
        <li>《明朝那些事儿》</li>
      </ul>

    </section>
    <blockquote>
      <p>“在人世间有一种庸俗势力的大合唱，谁一旦对它屈服，就永远沉沦了，真是可惜。”</p>
      <footer><cite>—王小波 《爱你就像爱生命》</cite></footer>
    </blockquote>
  </article>
</main>

<footer>
  <hr>
  <p>Written and coded by chaoqun.</p>
</footer>
</body>
</html>

```

部署静态服务器

要求是能够让所有人通过浏览器来访问。

简单地说，需要把刚刚写好的html文件放在一个有公网IP的服务器上，这样，无论是通过域名（已有域名并配置）还是ip地址，才能把http请求送到到正确的服务器上。

知道了地址还不行，还需要知道是哪一个应用程序才能正确的把信息送到并获取信息。我们是通过http协议来访问，所以默认就是80端口。我们需要一个服务器来监听80端口，从而在收到静态资源请求的时候返回相应的页面。

粗浅的回顾了原理以后，需要：

- 公网IP云服务器：主机
- web服务器：服务器软件

购买

按照要求，在腾讯云上租了一个按量计费的最低配（1 核 1 GB 1 Mbps）的服务器

这里一开始开通，不能直接拿代金券，原因应该是开通按量计费云服务器实例时，会预先冻结该CVM 一个小时的硬件费用（包括 CPU，内存，数据盘），所以我先冲了1块钱开通。最后结算的时候应该可以用代金券付款。

Nginx配置静态服务器

Nginx是一个Web服务器，相比Apache，内存占用少，稳定性好。当后期当前主机上有多个Web服务器是，还可用作反向代理。

```
# 登录云服务器：
ssh root@ip地址或者域名
# 然后输入密码

# pwd之后可以看到目前在 /root 目录
pwd

# 开始安装Nginx
# yum 是软件包管理器，下载安装并处理好依赖关系
sudo yum install yum-utils

# 进入/etc/yum.repos.d/
cd /etc/yum.repos.d/

# 新建 nginx.repo
```

```

vi nginx.repo
# 内容
[nginx-stable]
name=nginx stable repo
baseurl=http://nginx.org/packages/centos/$releasever/$basearch/
gpgcheck=1
enabled=1
gpgkey=https://nginx.org/keys/nginx_signing.key

[nginx-mainline]
name=nginx mainline repo
baseurl=http://nginx.org/packages/mainline/centos/$releasever/$basearch/
gpgcheck=1
enabled=0
gpgkey=https://nginx.org/keys/nginx_signing.key

# esc, :wq 修改保存之后, 安装
sudo yum install nginx
# 过程中会出现核实573B FD6B 3D8F BC64 1079 A6AB ABF5 BD82 7BD9 BF62, 一样就接受

# nginx是依赖configuration文件工作
# 默认情况下nginx.conf会出现在文件夹,
# /usr/local/nginx/conf, /etc/nginx, 或者 /usr/local/etc/nginx.

# 这里我这边是在/etc/nginx里面, 然后开启nginx
nginx -c /etc/nginx/nginx.conf

# 浏览器输入你购买的云服务器地址, 即可看到nginx启动成功页面
# 我们来看一下nginx.conf
cat /etc/nginx/nginx.conf

-----
user nginx;
worker_processes 1;

error_log /var/log/nginx/error.log warn;
pid /var/run/nginx.pid;

events {
    worker_connections 1024;
}

http {
    include /etc/nginx/mime.types;
    default_type application/octet-stream;

    log_format main '$remote_addr - $remote_user [$time_local] "$request" '

```

```

        '$status $body_bytes_sent "$http_referer" '
        '"$http_user_agent" "$http_x_forwarded_for"';

access_log /var/log/nginx/access.log main;

sendfile      on;
#tcp_nopush   on;

keepalive_timeout 65;

#gzip on;

include /etc/nginx/conf.d/*.conf;
}
-----

# 可以看到很多细节的配置, 这里只需要关注http{}内, include /etc/nginx/conf.d/*.conf;
# 这里导入了这个文件, 继续去看看

vi /etc/nginx/conf.d/default.conf

# 下面是我修改之后的文件, 把对 / 的请求映射到我们存放html的目录上
-----
server{
    listen      80;
    server_name localhost;

    #charset koi8-r;
    #access_log /var/log/nginx/host.access.log main;

    #location / {
    #    root /usr/share/nginx/html;
    #    index index.html index.htm;
    #}

    location / {
        root /data/www;
    }

    #error_page 404 /404.html;

    # redirect server error pages to the static page /50x.html
    #
    error_page 500 502 503 504 /50x.html;
    location = /50x.html {
        root /usr/share/nginx/html;
    }

    # proxy the PHP scripts to Apache listening on 127.0.0.1:80

```

```
#
#location ~ /\.php$ {
#    proxy_pass    http://127.0.0.1;
#}

# pass the PHP scripts to FastCGI server listening on 127.0.0.1:9000
#
#location ~ /\.php$ {
#    root          html;
#    fastcgi_pass  127.0.0.1:9000;
#    fastcgi_index index.php;
#    fastcgi_param SCRIPT_FILENAME /scripts$fastcgi_script_name;
#    include       fastcgi_params;
#}

# deny access to .htaccess files, if Apache's document root
# concurs with nginx's one
#
#location ~ /\.ht {
#    deny  all;
#}
}

-----

# 新开一个shell, 把你写好的html文件上传到该服务器上
scp 你的html文件地址 云服务器放文件的地址
# 例:  scp html文件地址 root@ip地址:/data/www/index.html
# 前提你已经建好了 /data/www/ 这个目录

# 重新加载配置文件, 不中断服务器
nginx -s reload

# 刷新浏览器可以看到你的个人介绍页面
```

我的页面: [个人页面](#)

参考链接

[Nginx](#)

[nginx: Linux packages](#)

[计费模式](#)

[文档与网站架构](#)