参考：<http://blog.csdn.net/ozhengtuojiasuo/article/details/52803646>

**location 语法**

location 有”定位”的意思, 根据Uri来进行不同的定位.

在虚拟主机的配置中,是必不可少的,location可以把网站的不同部分,定位到不同的处理方式上.

比如, 碰到.php, 如何调用PHP解释器?  --这时就需要location

location 的语法

location [=|~|~\*|^~] patt {

}

=:等于匹配，location后面的与url必须完全匹配才能将访问定位到此处理

~: 普通匹配，匹配路径，不包含host

~\*:与~类似，对路径不区分大小写

^~：类似于正则规则，匹配开头

中括号可以不写任何参数,此时称为一般匹配

也可以写参数

因此,大类型可以分为3种

location = patt {} [精准匹配]

location patt{}  [一般匹配]

location ~ patt{} [正则匹配]

location ~\*patt{} [正则匹配，不区分大小写]

**如何发挥作用?**

首先看有没有精准匹配,如果有,则停止匹配过程.

location = patt {

   config A

}

如果 $uri ==patt,匹配成功，使用configA

  location = / {

            root   /var/www/html/;//只在本处理区域中适用

            index  index.htm index.html;

       }

如果访问　　[http://xxx.com](http://xxx.com/)/

定位流程是

1: 精准匹配中　”/”   ,得到index页为　　index.htm

2: 再次访问 /index.htm , 此次内部转跳uri已经是”/index.htm” ,

根目录为/usr/local/nginx/html，

在处理规则中定义的变量如root只适用于此处理规则，对其他得处理规则不适用，如果需要全局适用需要将root定义到规则的外面http或者server中。

server {}或者http{}

3: 最终结果,访问了/usr/local/nginx/html/index.htm

再来看,正则也来参与.

location / {//一般匹配

            root   /usr/local/nginx/html;

            index  index.html index.htm;

        }

location ~ image {//正则匹配

           root /var/www/image;

           index index.html;

}

如果我们访问  <http://xx.com/image/logo.png>

此时, “/”与”/image/logo.png”匹配

同时,”image”正则与”image/logo.png”也能匹配,谁发挥作用?

正则表达式的成果将会使用，更精确地发挥作用

图片真正会访问 /var/www/image/logo.png

location / {

            root   /usr/local/nginx/html;

            index  index.html index.htm;

        }

location /foo {

           root /var/www/html;

            index index.html;

}

我们访问http://xxx.com/foo

 对于uri“/foo”,   两个location的patt,都能匹配他们

即‘/’能从左前缀匹配‘/foo’, ‘/foo’也能左前缀匹配’/foo’,

此时, 真正访问 /var/www/html/index.html

原因:’/foo’匹配的更长,因此使用之.;