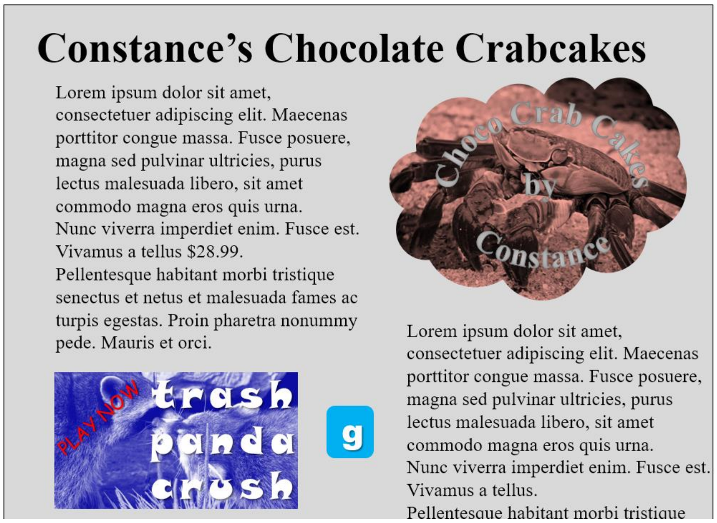
# Web Example

## Web page

Let’s suppose the following web page is at www.crabcandy.com/chococrabcakes.html[1].

This web page has text, their own logo (upper right), an ad for the hit game Trash Panda Crush (lower left) and the logo of well known social media site goosebook (middle).

Let’s say that on this computer[2] you’ve never visited this site before, but you have seen the goosebook logo on other pages. The Trash Panda Crush ad is stored on the server of www.racoonads.com, and you’ve never visited a page with one of their ads before.



Constance’s Chocolate Crabcakes

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus $28.99. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Proin pharetra nonummy pede. Mauris et orci.

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Maecenas porttitor congue massa. Fusce posuere, magna sed pulvinar ultricies, purus lectus malesuada libero, sit amet commodo magna eros quis urna. Nunc viverra imperdiet enim. Fusce est. Vivamus a tellus. Pellentesque habitant morbi tristique

To visit this page, you type in the URL to your browser.

[1] All URLs used here were not in live use at the time they were made up for this example.
[2] Whether you’ve done so on other computers or not won’t affect this example.

# DNS Lookup

Since you've never visited the page before, your browser has to do a DNS lookup for www.crabcandy.com, and sends a request to your nearest Domain Name Server. Since this is not a widely known site, your request will probably bounce through a few DNS servers before one of them finds www.crabcandy.com in its table and sends back an IP address, say 13.24.57.68[3].

Now this IP address is stored for later in your computer, and the browser can actually start getting the page.

# Getting the Web Page

The browser sends the GET command to request the web page.

So it sends to 13.24.57.68:80 the message

```
GET http://www.crabcandy.com/ chococrabcakes.html
```

This message arrives at the 102chocolatecrabcakes.com webserver, which looks for the requested file, chococrabcakes.html. It sends this web page back to your computer.

# Rendering the Web Page

The web page is written in html and consists of text and formatting codes, but none of the pictures. Your browser starts rendering this code to display the page.

[3] IIP addresses were chosen for this example purely for memorable numbers.

## First Image – same server

When the browser reaches the first <img> tag, it sees the URL for the first image: http://www.crabcandy.com/logo.jpg.

So, it must send a new GET command.  Since this URL is also on the crabcandy.com webserver, it doesn't need to do a new DNS lookup, it just sends to 13.24.57.68:80 the message

```
GET http://www.crabcandy.com/logo.jpg
```

This message arrives at the www.crabcandy.com webserver, which looks for the requested file, logo.jpg.  It sends this image file back to your computer, and the browser displays it on screen.

## Second image – ad server

In the meantime, the browser has continued rendering the page, and found another <img> tag with the URL http://www.racoonads.com/crabcandy_ad1.gif for the ad so it needs to send a GET for this file.

Since this file is at a different server address, and we assumed you had never seen an ad from that company before, a new DNS lookup has to happen.  Your computer sends a DNS lookup request for www.racoonads.com and eventually gets the IP address back, say 11.22.33.44.  This is stored for later, and a GET can now be sent to 11.22.33.44:80

```
GET http://www.racoonads.com/crabcandy_ad1.gif
```

This message arrives at the www.racoonads.com webserver, which looks for the requested file, crabcandy_ad1.gif.

The ad server sends an ad image file back to your computer. Various companies have paid this advertising service to show their ad images on web pages.  Since at this point the ad service has no information about you, it will just pick an ad image, perhaps based on which company paid for their image to be shown most often, or other ads listed as being related to crabs or candy.

In addition to the ad image, it also sends a cookie. Because of the image requested, it knows which site you were looking at, so it knows the date and time you viewed this site, and this information is stored in the cookie.

```
cookie: www.racoonads.com
updated 01/02/2020 11:23
01/02/2020 11:23 viewed www.crabcandy.com
```

Your browser displays the image and stores the cookie for later.

Since this cookie is for a server other than the one for the page you were visiting, it is a *third party cookie*.  If www.crabcandy.com had used a cookie, that would be a first party cookie, but in this example that didn't happen.

## Third image – goosebook

While the other images were being handled the browser continued rendering the page, and found another <img> tag with the URL http://www.goosebook.com/crabcandy_logo.gif for the goosebook logo, so it needs to send a GET for this file as well.

We assumed you had seen goosebook's logo before, so a new DNS lookup is not necessary, your computer must have done that lookup sometime earlier and stored the IP address, say 44.33.22.11.

Since you have communicated with the www.goosebook.com web server before, you probably have a cookie for that server stored on your computer, which might have information stored in it about other sites they know you viewed because you requested the goosebook logo on those sites.  The cookie might look like:

```
cookie: www.goosebook.com
updated 22/01/2020 12:12
11/12/2019 10:41 viewed mail.somanygoslings.gx
30/12/2019 05:02 viewed www.harford.edu
22/01/2020 12:12 viewed www.grouses4spouses.co.uk
```

So, when your browser sends the GET, it includes this cookie along with the URL it is requesting

```
GET http://www.goosebook.com/crabcandy_logo.gif
cookie: www.goosebook.com
updated 22/01/2020 12:12
11/12/2019 10:41 viewed mail.somanygoslings.gx
30/12/2019 05:02 viewed www.harford.edu
22/01/2020 12:12 viewed www.grouses4spouses.co.uk
```

This message arrives at the www.goosebook.com webserver, which looks for the requested file, crabcandy_logo.gif.

The goosebook server sends the logo image file back to your computer, but it also updates the cookie and sends it back. Because of the image requested, it knows which site you were looking at, so it knows the date and time you viewed this site, and this information is added to this third party cookie.

```
cookie: www.goosebook.com
updated 01/02/2020 11:23
11/12/2019 10:41 viewed mail.somanygoslings.gx
30/12/2019 05:02 viewed www.harford.edu
22/01/2020 12:12 viewed www.grouses4spouses.co.uk
01/02/2020 11:23 viewed www.crabcandy.com
```

Your browser displays the image and stores the cookie for stored for the next time you talk to this server

## Later

Suppose that the next day on the same computer you visit another site that uses an ad from the www.racoonads.com server. Then the cookie your browser stored for that server will be sent along with the GET you send for the ad image. This allows the server to update the cookie with your visit to this new site.

It also means that this time the ad server has some information about you that it can use to choose which ad image to send to you[4]. You may be browsing a site on waterskis, but see an ad related to crabs or candy.

However, if you were to visit a site with a www.racoonads.com ad on it from another computer, there would be a totally different cookie, listing only sites you had visited from that computer, and you would only get crab or candy ads by chance, not because they knew it was among your interests.

Next, back on the first computer, you login to goosebook. Your GET command when logging in includes the URL, your username and password, and the current cookie. Based on this, the server sends back the page, updates the cookie with the information that you are currently logged in (with a timestamp). In this case, since you were visiting the goosebook page, this is a first party cookie.

[4] Such sites do also usually collect such data in a huge store of information without storing the IP addresses it came from. They use this to try to find overall trends, such as the percentage of people interested in both crabs and waterskiing.

The goosebook.com server will probably also pull the information about the sites you visited with goosebook logos out of the cookie and store it in your account.  Now goosebook[5] might show you ads for crabs or candy even on other computers when you are logged in, because they have your interest in this stored on the server, not just inside a cookie.

[5] Goosebook will also certainly use your data to try to find overall trends, and goosebook and racoonads may certainly sell this trend information to each other.