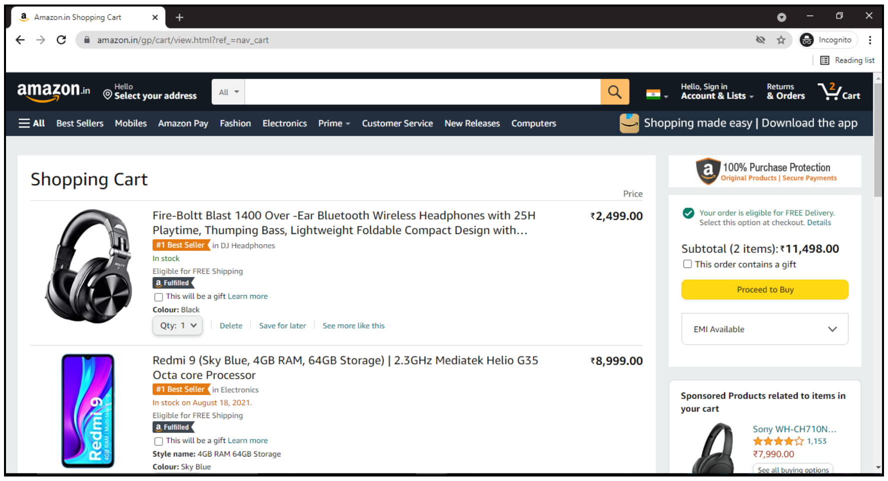
Day - 10
Introduction to Cookies

In this session we will discuss **Cookies**. Cookies allows the web applications to provide a personalized experience for the user using their websites.
Cookies are the piece of information that the server stores inside the web browser and reuses that information to provide personalized user experience for the user.
Let us take an example of **Amazon**.
When we add items to our virtual cart and if we close the browser, then again if we launch amazon, the items would still present in the cart even though we had not signed in.
This is possible because of the cookies.



Shopping Cart

Price

Fire-Boltt Blast 1400 Over -Ear Bluetooth Wireless Headphones with 25H Playtime, Thumping Bass, Lightweight Foldable Compact Design with... ₹2,499.00
#1 Best Seller in DJ Headphones
In stock
Eligible for FREE Shipping
Fulfilled
This will be a gift Learn more
Colour: Black
Qty: 1 | Delete | Save for later | See more like this

Redmi 9 (Sky Blue, 4GB RAM, 64GB Storage) | 2.3GHz Mediatek Helio G35 Octa core Processor ₹8,999.00
#1 Best Seller in Electronics
In stock on August 18, 2021.
Eligible for FREE Shipping
Fulfilled
This will be a gift Learn more
Style name: 4GB RAM 64GB Storage
Colour: Sky Blue

Your order is eligible for FREE Delivery.
Select this option at checkout. Details
Subtotal (2 items): ₹11,498.00
This order contains a gift

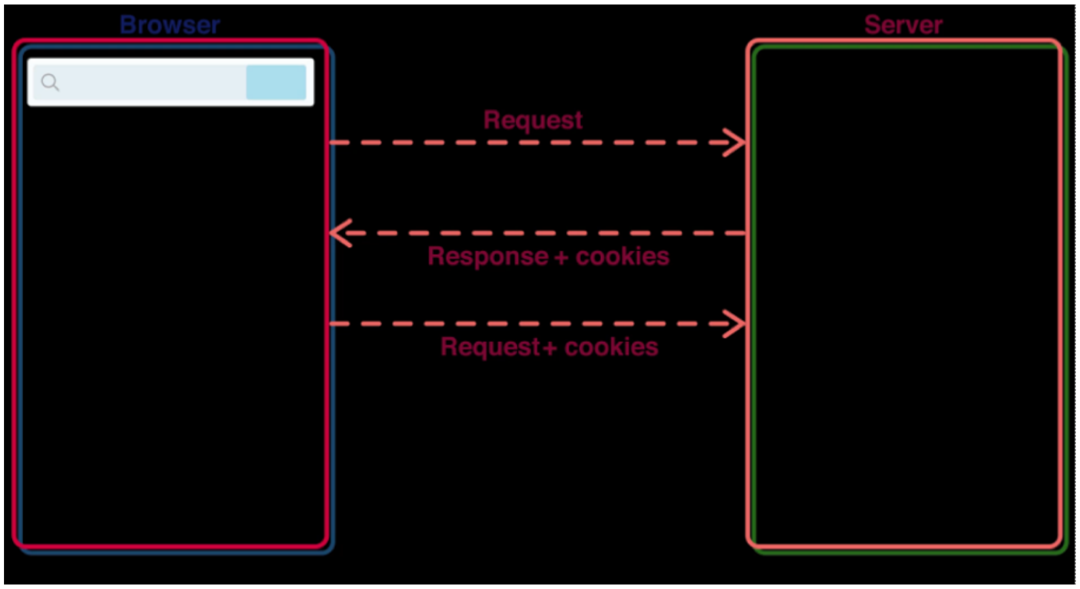
Sponsored Products related to items in your cart
Sony WH-CH710N...
1,153
₹7,990.00
See all buying options

We can consider another example, where whenever we visit any websites, there will be ads on the top and sides and these ads are called Banner Ads.
So next time when we revisit the same website, the ads would be relevant to what we had previously searched for, because the website would have collected the information and stored it in the form of cookies inside the browser.
Cookies can be classified into two categories-

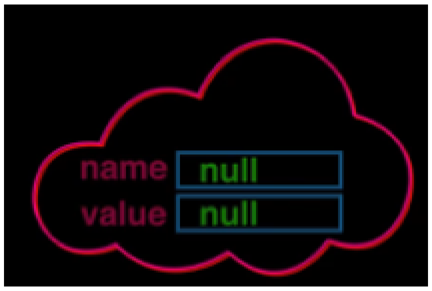
- ➔ First party cookies
- ➔ Third party cookies

Let us now try to understand how cookies work-

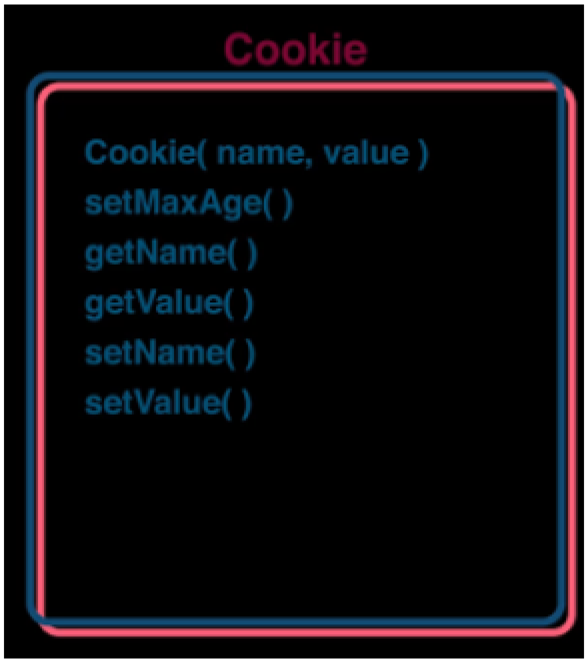
The first time the browser sends a request to a server, the server will internally create a cookie and it sends the cookie along with the response
Next time when we send the request from the browser to the server, it will also attach the cookie with the request and that is what the server reads and understands the information about the user.



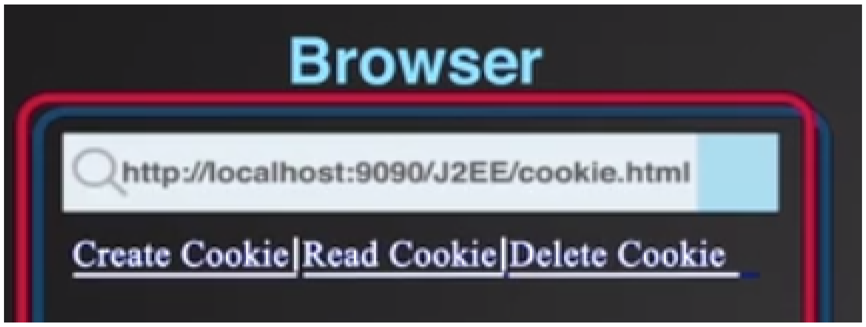
But how to create cookies?
Since Java is an object-oriented programming language, everything is considered as an object and we have a class called **Cookie** in java.
Whenever we want to create a cookie, we have to create an object of the **Cookie class** and inside this object, there will be two instance variables, **name, and value** which are both of **String** type.



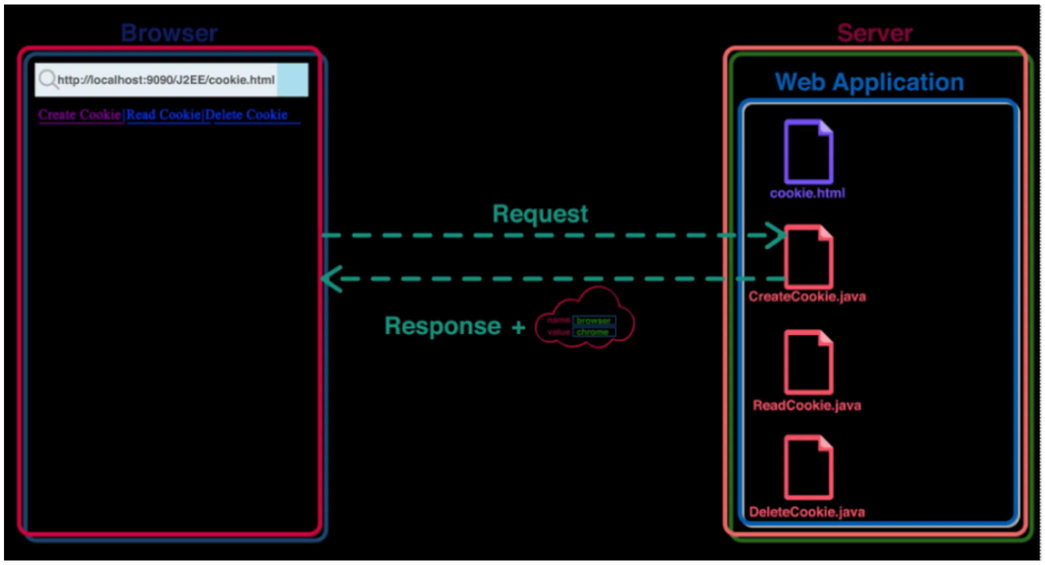
There are a variety of methods inside the Cookie class as shown below



Now let us create our first cookie program and the program expectation is as follows-
In our JEE project, we will create an HTML file as cookie.html and 3 servlets **CreateCookie.java**, **ReadCookie.java,** and **DeleteCookie.java.**
When a user types in the URL "http://localhost:9090/J2EE/cookie.html" and presses enter, now the server will send, cookie.html as a response, and in the browser, a navigation bar containing 3 hyperlinks should be displayed as shown-


Browser
http://localhost:9090/J2EE/cookie.html
Create Cookie|Read Cookie|Delete Cookie

Now if the user clicks on the ***Create Cookie link,*** then a request from the browser should be sent to the **CreateCookie.java**.
When the request is sent to the CreateCookie servlet for the first time, the CreateCookie servlet will create a cookie object with the help of the Cookie class.
Now the CreateCookie servlet will send the cookie along with the response



The cookie will reach the browser and the browser will store the cookie in its internal memory and cookies will have a lifetime of 90 secs and this time, which is the maximum age the cookie can live in a browser, can be set by making use of a method called setMaxAge(), and as soon as the time set is up, automatically the cookie will be deleted from the browser.
Let us now try to implement this in our code-

1. Create a package **"com.tap.cookie"**
2. Create an HTML file by right-clicking on the **project** → **New** → **HTML file** and set the name as **cookie.html**
   a. Type in the following code in the HTML file

```
<!DOCTYPE html>
```

```html
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <a href="">Create cookie</a><span>|</span>
      <a href="">Read cookie</a><span>|</span>
      <a href="">Delete cookie</a>
</body>
</html>
```

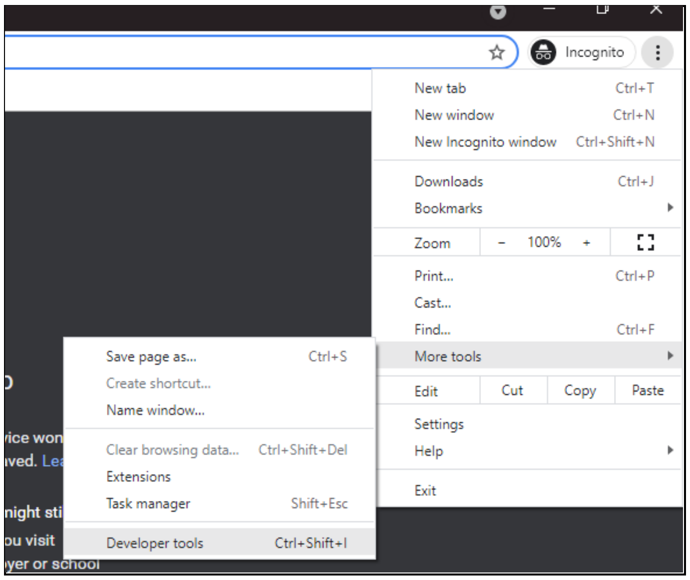
2. Create **“CreateCookie.java”** servlet in the **“com.tap.cookie”** package and type in the following-

```java
package com.tap.cookie;

@WebServlet("/CreateCookie")
public class CreateCookie extends HttpServlet {

      @Override
      protected void doGet(HttpServletRequest req, HttpServletResponse resp)
            throws ServletException, IOException {
            resp.setContentType("text/html");
            PrintWriter writer = resp.getWriter();

            Cookie ck = new Cookie("browser", "chrome");
            ck.setMaxAge(90);
            resp.addCookie(ck);
            writer.println("<h3>Cookie has been created</h3>");
      }

      }
```

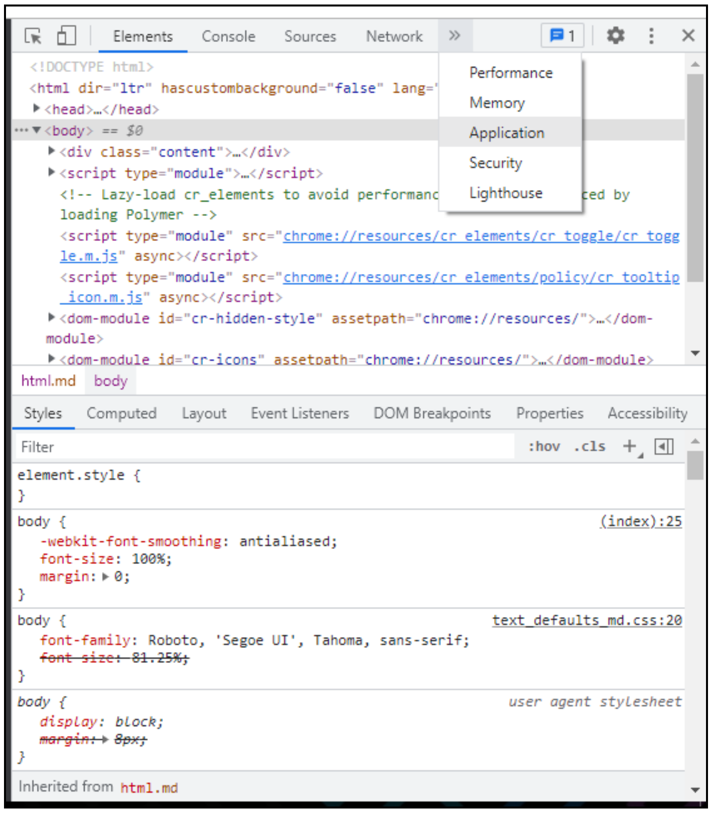
3. Add the Relative URL in the anchor tag in the cookie.html file

```html
<a href="CreateCookie">Create cookie</a><span>|</span>
```

Every browser gives us the ability to see the cookies that have been saved by the browser.
Open **Developer tools**, by clicking on the 3 dots present at the top right corner of the browser and select **more tools → Developer tools**



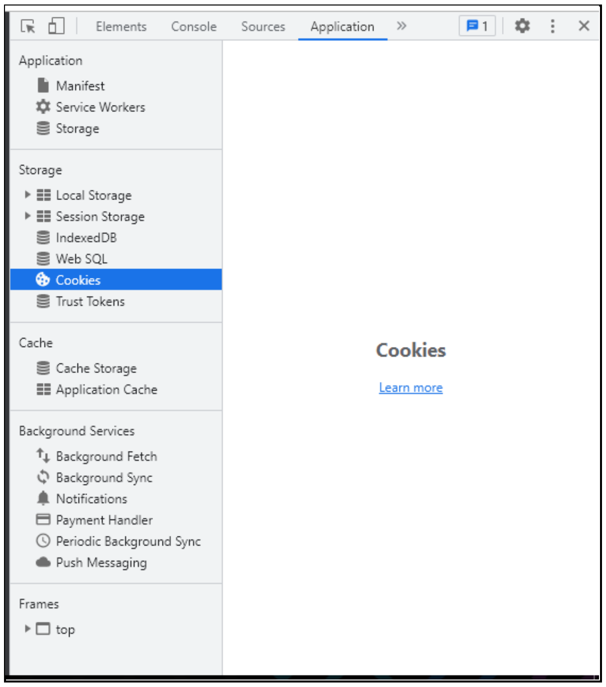
Now click on the **Application** tab



```
<!DOCTYPE html>
<html dir="ltr" hascustombackground="false" lang="
 <head>…</head>
••• <body> == $0
   <div class="content">…</div>
   <script type="module">…</script>
    <!-- Lazy-load cr_elements to avoid performanc        ced by
    loading Polymer -->
    <script type="module" src="chrome://resources/cr_elements/cr_toggle/cr_toggle.m.js" async></script>
    <script type="module" src="chrome://resources/cr_elements/policy/cr_tooltip_icon.m.js" async></script>
   <dom-module id="cr-hidden-style" assetpath="chrome://resources/">…</dom-module>
   <dom-module id="cr-icons" assetpath="chrome://resources/">…</dom-module>
```

```
element.style {
}
body {                                                  (index):25
   -webkit-font-smoothing: antialiased;
   font-size: 100%;
   margin: ▸ 0;
}
body {                                      text_defaults_md.css:20
   font-family: Roboto, 'Segoe UI', Tahoma, sans-serif;
   ~~font-size: 81.25%;~~
}
body {                                      user agent stylesheet
   display: block;
   ~~margin: ▸ 8px;~~
}
```

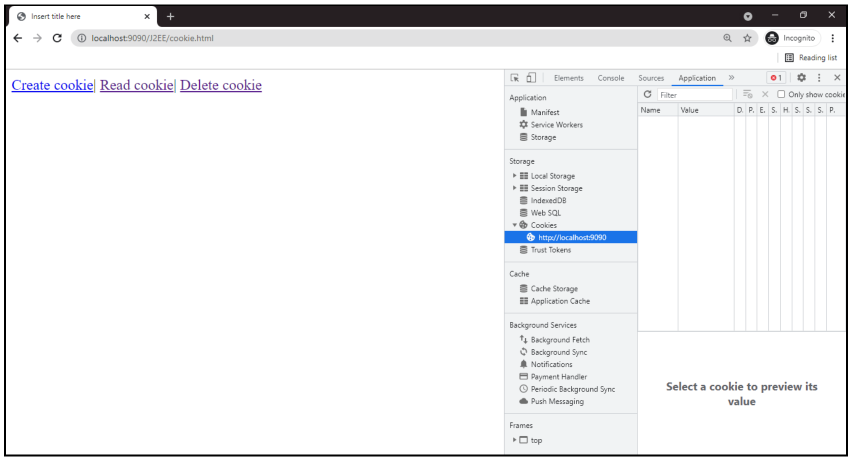
and click on the Cookies tab



Cookies

Learn more

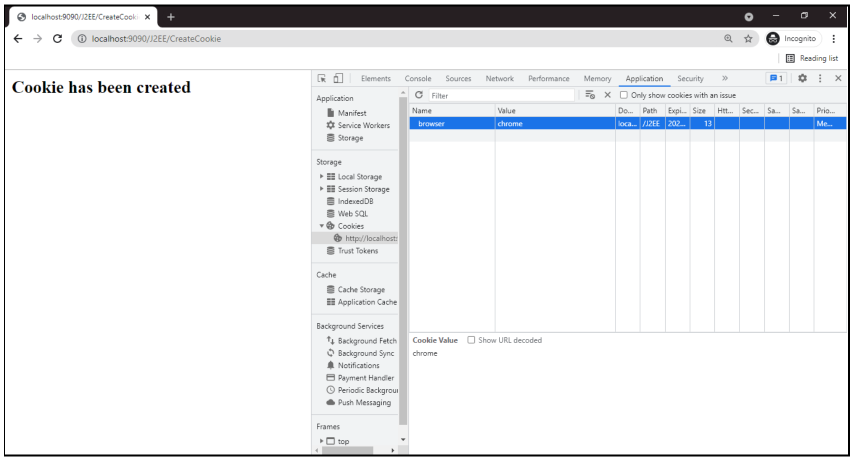
And when we request for cookie.html file in the browser by typing "http://localhost:9090/J2EE/cookie.html", we will get the following output-



Create cookie| Read cookie| Delete cookie

Select a cookie to preview its value

And when we click on Create cookie link, then we can see that cookie would have been added to our browser


localhost:9090/J2EE/CreateCooki
localhost:9090/J2EE/CreateCookie
Incognito
Reading list
Elements
Console
Sources
Network
Performance
Memory
Application
Security
Filter
Only show cookies with an issue

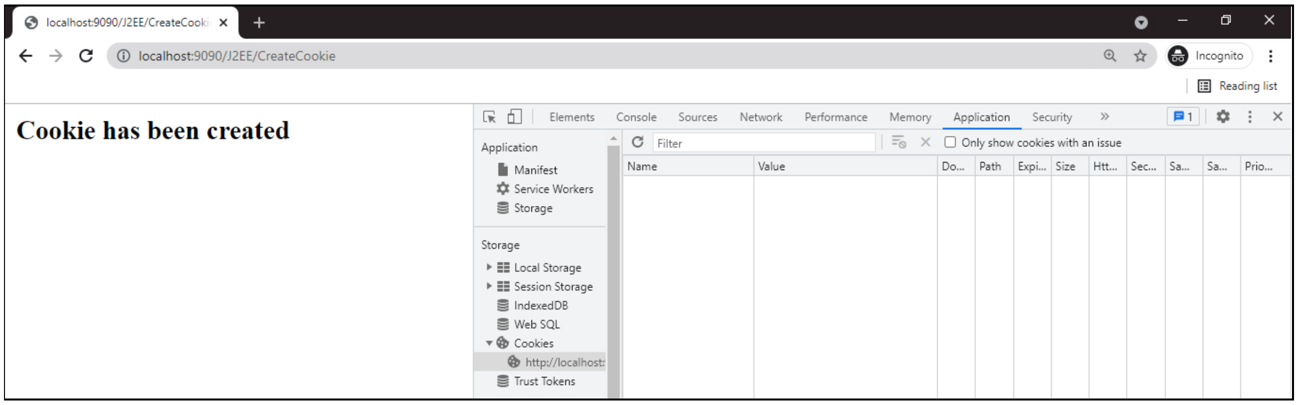
Cookie has been created

Name	Value	Do...	Path	Expi...	Size	Htt...	Sec...	Sa...	Sa...	Prio...
browser	chrome	loca...	/J2EE	202...	13					Me...

Cookie Value ☐ Show URL decoded

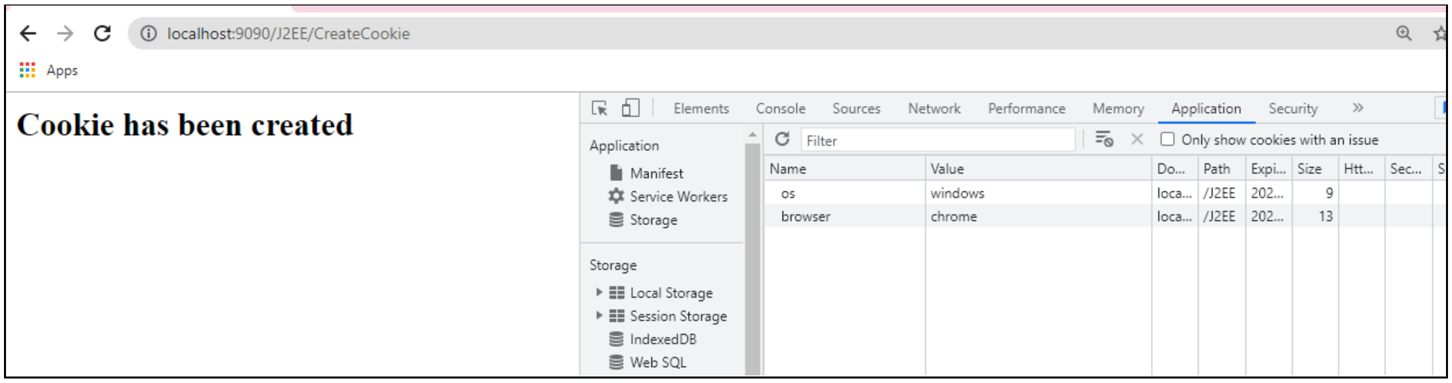
chrome

As soon as 90 secs are up, the cookie will be automatically deleted from the browser.


Cookie has been created

We can create as many cookies as we want

```
Cookie ck1 = new Cookie("browser", "chrome");
ck1.setMaxAge(90);
resp.addCookie(ck2);

Cookie ck2 = new Cookie("os", "windows");
ck2.setMaxAge(90);
resp.addCookie(ck2);
```


Cookie has been created

Name	Value	Do...	Path	Expi...	Size	Htt...	Sec...
os	windows	loca...	/J2EE	202...	9		
browser	chrome	loca...	/J2EE	202...	13		

Now that we have understood how to create a cookie, let us now see how to read a cookie

When a user clicks on the **Read Cookie** link, then along with the Request, a cookie will also be sent to the browser

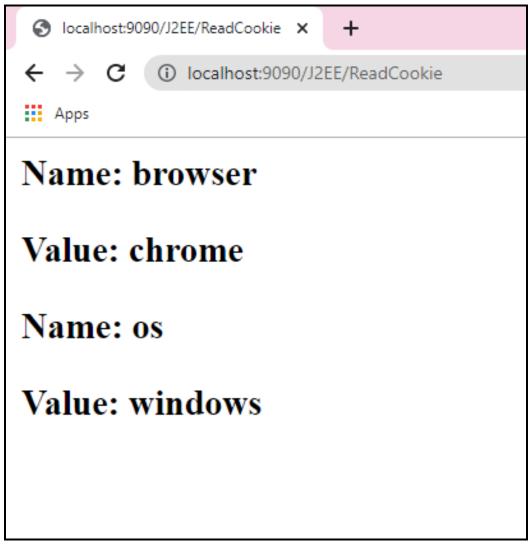
1. Create a **ReadCookie.java** servlet in *com.tap.cookie* package and type in the following-

```
package com.tap.cookie;

@WebServlet("/ReadCookie")
public class ReadCookie extends HttpServlet {

      @Override
      protected void doGet(HttpServletRequest req, HttpServletResponse resp)
                  throws ServletException, IOException {

            resp.setContentType("text/html");
            PrintWriter writer = resp.getWriter();

            Cookie[] cookies = req.getCookies();

            for(int i=0;i<cookies.length;i++)
            {
                  String name = cookies[i].getName();
                  String value = cookies[i].getValue();
                  writer.println("<h3>Name: "+name+"</h3>");
                  writer.println("<h3>Value: "+value+"</h3>");
            }
      }
}
```

2. Map the servlet in cookie.html file as shown-

```
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <a href="CreateCookie">Create cookie</a><span>|</span>
      <a href="ReadCookie">Read cookie</a><span>|</span>
      <a href="">Delete cookie</a>
</body>
</html>
```

And if we execute this in our browser, then we will get the following output-

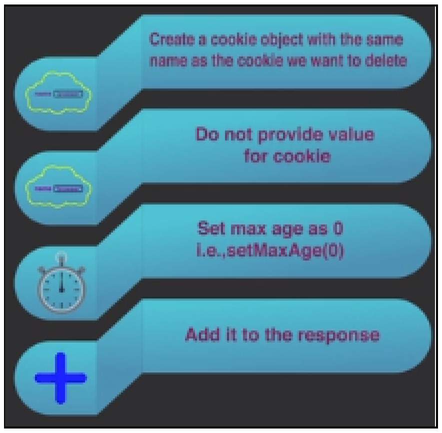

localhost:9090/J2EE/ReadCookie
localhost:9090/J2EE/ReadCookie
Apps

```
Name: browser

Value: chrome

Name: os

Value: windows
```

Now let us see how to delete a cookie-

Whenever we want to delete a cookie, we have to follow these steps-

- ➔ Create a cookie object with the same name as the cookie we want to delete
- ➔ Do not provide value for the cookie
- ➔ Set max age as 0 i.e., setMaxAge(0)
- ➔ Add it to the response


Create a cookie object with the same name as the cookie we want to delete
Do not provide value for cookie
Set max age as 0 i.e.,setMaxAge(0)
Add it to the response

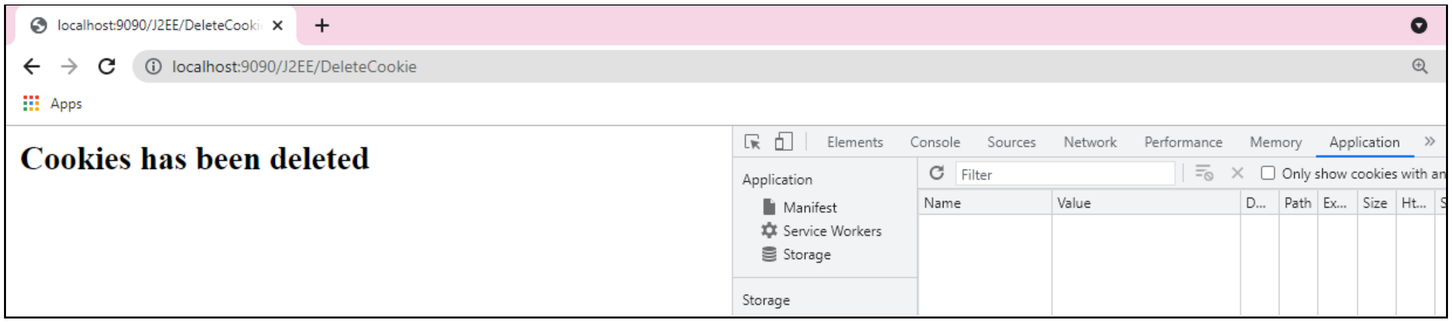
1. Create a **DeleteCookie.java** servlet in *com.tap.cookie* package and type in the following-

```
package com.tap.cookie;

@WebServlet("/DeleteCookie")
public class DeleteCookie extends HttpServlet {

      @Override
      protected void doGet(HttpServletRequest req, HttpServletResponse resp)
                  throws ServletException, IOException {

            resp.setContentType("text/html");
            PrintWriter writer = resp.getWriter();

            Cookie ck1 = new Cookie("browser", "");
            ck1.setMaxAge(0);
            resp.addCookie(ck1);

            Cookie ck2 = new Cookie("os", "");
            ck2.setMaxAge(0);
            resp.addCookie(ck2);

            writer.println("<h3>Cookies has been deleted</h3>");
      }
}
```

2. Map the servlet in cookie.html file as shown-

```html
<!DOCTYPE html>
<html>
<head>
<meta charset="ISO-8859-1">
<title>Insert title here</title>
</head>
<body>
      <a href="CreateCookie">Create cookie</a><span>|</span>
      <a href="ReadCookie">Read cookie</a><span>|</span>
      <a href="DeleteCookie">Delete cookie</a>
</body>
</html>
```

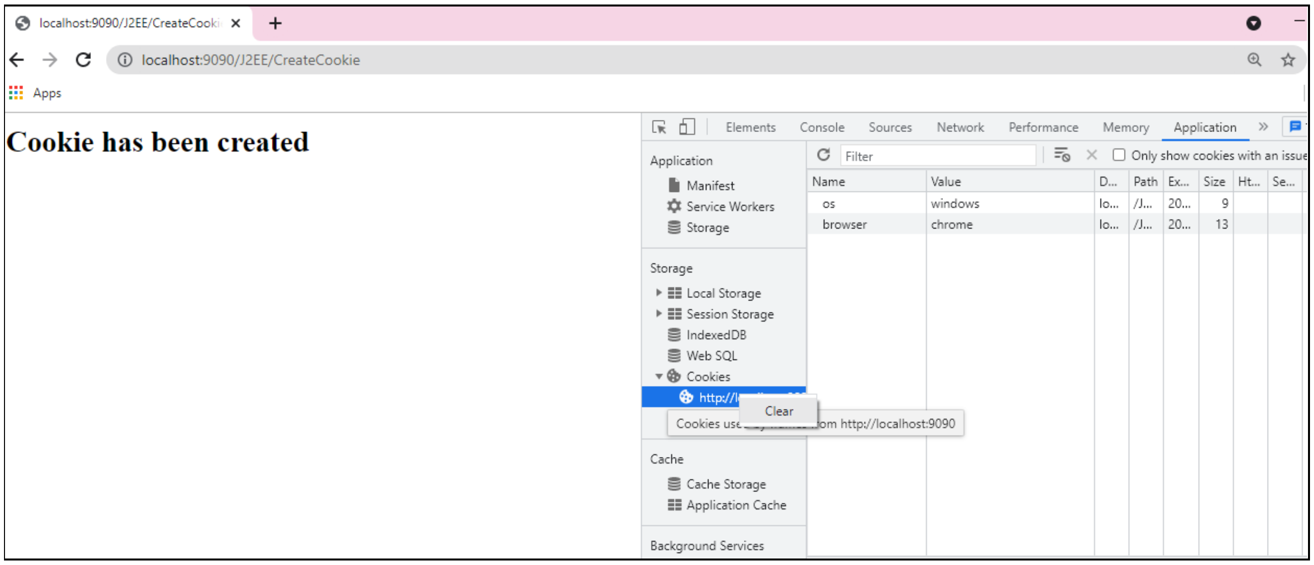
Upon execution, we can see that the cookies have been deleted.


localhost:9090/J2EE/DeleteCookie

Cookies has been deleted

Name	Value	D...	Path	Ex...	Size	Ht...

But if you want to delete a cookie manually, then all we have to do is right-click on the Cookies tab in the browser and then clicking on the clear button as shown below-


localhost:9090/J2EE/CreateCookie

Cookie has been created

Name	Value	D...	Path	Ex...	Size	Ht...	Se...
os	windows	lo...	/J...	20...	9		
browser	chrome	lo...	/J...	20...	13		

Even though cookies are useful, it has its disadvantages,

Disadvantages of cookies-

➔ Cookies can be disabled by the browsers

If cookies are disabled, then we will no longer be able to add items to the cart and login to our mail id as the browser will no longer be able to store/remember the password as cookies have been disabled