Universidad Nacional
de La Matanza

DIIT
Departamento de Ingeniería e
Investigaciones Tecnológicas

Tecnicatura en Desarrollo Web- Aplicaciones Móviles
INGLÉS TÉCNICO 2 - cód 2627/ 3003

KEY UNIT 4

Pag 43-44

En el siguiente texto elija los conectores correctos.Luego complete el cuadro

What is Ajax?

Ajax (sometimes called Asynchronous JavaScript and XML) is a way of programming for the Web that gets rid of the hourglass. Data, content, and design are merged together into a seamless whole. ***Before // After*** *your customer clicks on something on an Ajax driven application, there is very little lag time. The page simply displays what they're asking for.* ***If // therefore*** *you don't believe me, try out Google Maps for a few seconds. Scroll around and watch as the map updates almost before your eyes. There is very little lag* ***; therefore, //because*** *you don't have to wait for pages to refresh or reload.*

Ajax is a way of developing Web applications that combines:

_ XHTML and CSS standards based
presentation _ Interaction with the page
through the DOM _ Data interchange with
XML and XSLT

_ Asynchronous data retrieval with
XMLHttpRequest _ JavaScript to tie it all
together

In the traditional Web application, the interaction between the customer and the server goes like this:

_ Customer accesses Web application

*_ Server **either** processes the request **or** sends **//** **not only** processes the request **but also** sends data to the browser while the customer waits*

_ Customer clicks on a link or interacts with the application

*_ Server processes request and sends data back to the browser **then // while** the customer waits _ etc....*

***However, /// As a result,** there is a lot of customer waiting.*

Ajax Acts as an Intermediary

*The Ajax engine works within the Web browser (through JavaScript and the DOM) to render the Web application and handle any requests that the customer might have of the Web server. **As // After** the Ajax engine is handling the requests, it can hold most information in the engine itself **while /although** allowing the interaction with the application and the customer to happen asynchronously and independently of any interaction with the server.*

Asynchronous

*This is the key. In standard Web applications, the interaction between the customer and the server is synchronous. This means that one has to happen after the other. **Due to // If** a customer clicks a link, the request is sent to the server, which then sends the results back. With Ajax, the JavaScript that is loaded when the page handles most of the basic tasks such as data validation and manipulation **as well as // moreover** display rendering without a trip to the server. **While // During** it is making display changes for the customer, it is sending data back and forth to the server. **Although, / However,** the data transfer is not dependent upon actions of the customer.*

Adapted from: http://webdesign.about.com/od/ajax/a/aa101705.htm

Conjunción / Nexo conector	Fragmentos que une	Relación /Función
After	*your customer clicks on something on an Ajax driven application*	TIME / TIEMPO
	there is very little lag time	
If	*you don't believe me*	CONDITION/ CONDICIÓN
	try out Google Maps for a few seconds.	
therefore	*There is very little lag.*	RESULT/ RESULTADO
	you don't have to wait for pages to refresh or reload	
not only	*processes the request*	correlación
but also	*sends data to the browser while…the customer waits*	
As a result	while the customer waits	RESULT/ RESULTADO
	there is a lot of customer waiting	
As	*the Ajax engine is handling the requests*	TIME / TIEMPO
	it can hold most information in the engine itself	
while	*it can hold most information in the engine itself*	TIME / TIEMPO
	allowing the interaction with the application and the customer to happen asynchronously and independently of any interaction with the server	
If	*a customer clicks a link*	CONDITION/ CONDICIÓN
	the request is sent to the server	
as well as	*the page handles most of the basic tasks such as data validation and manipulation*	Addition/ ADICIÓN
	display rendering without a trip to the server	

While	*it is making display changes for the customer*	TIME / TIEMPO
	it is sending data back and forth to the server	
However	*it is sending data back and forth to the server*	CONTRAST/ CONTRAST
	the data transfer is not dependent upon actions of the customer.	