

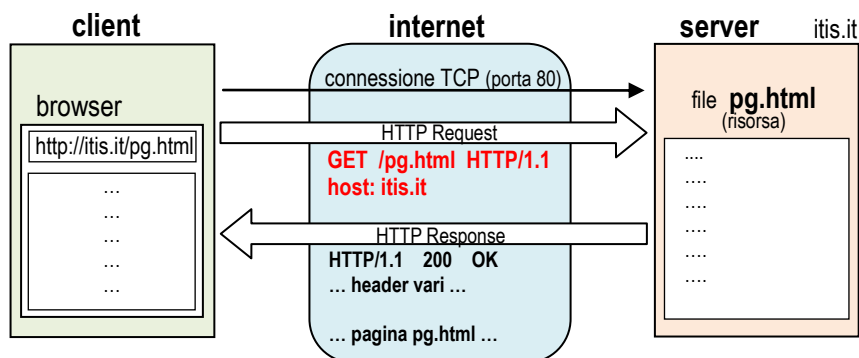
## Protocollo HTTP: Invio di Dati al Server Web

<p style="text-align: center;"><b>Come è Definito l'URL con Parametri</b></p>	<p>E' possibile <i>inviare dei dati</i> a un server web, inserendoli direttamente nell'<i>URL</i>. Si tratta del cosiddetto <b>invio di parametri nell'URL</b>.</p> <p style="text-align: center;"><i>&lt;protocollo&gt; :// &lt;dominio&gt; / &lt;risorsa&gt; ? &lt;parametri&gt;</i></p> <p>Esempi:  <b>http://www.supermercato.it/offerte.php?mese=3&amp;giorno=15</b>  <b>http://www.negozio.it/schede.php?prodotto=A50</b></p> <hr/> <p><b>&lt;parametri&gt;</b>          L'elenco dei parametri, ossia dei dati, da trasmettere al server web.          Nell'URL, la <i>&lt;risorsa&gt;</i> è separata dai <i>&lt;parametri&gt;</i>, dal simbolo <b>?</b> (<b>Punto Interrogativo</b>)          Un <b>Parametro</b> è definito da un <b>Nome</b> e da un <b>Valore</b>.          E' possibile inviare <i>più di un parametro</i> secondo la regola:  <i>&lt;nome&gt;=&lt;valore&gt;&amp;&lt;nome&gt;=&lt;valore&gt;&amp; ...</i>          Questo formato è indicato con il nome <b>application/x-www-form-urlencoded</b></p>
<p style="text-align: center;"><b>Come il Server Recupera i Parametri dall'URL (PHP)</b></p>	<p>Quando si inviano parametri nell'URL, la risorsa richiesta generalmente è una <b>pagina web dinamica</b> che prevede l'<b>esecuzione di codice sul server</b> (PHP o ASPX o JSP, ecc.)          Il codice della pagina deve "recuperare" il valore dei parametri. In linguaggio PHP si utilizza la variabile super-globale (cioè, sempre disponibile) chiamata <b>\$_GET</b> :</p> <p><b>\$_GET [ '&lt;nome-parametro&gt;' ]</b> ...restituisce il <b>valore</b> del parametro specificato</p> <p>Esempio:  <b>\$ValoreDelMese = (int) \$_GET [ 'mese' ];</b>      ... \$ValoreDelMese ora vale 3...  <b>\$ValoreDelGiorno = (int) \$_GET [ 'giorno' ];</b>      ... \$ValoreDelGiorno ora vale 15...</p> <p>Il valore dei parametri è restituito sempre come valore di <b>tipo string</b>, quindi, se si tratta di un dato numerico è <b>necessaria una conversione</b>.          Per conoscere il <b>numero di parametri disponibili</b>, si usa la funzione <b>count( )</b></p> <p>Esempio:  <b>if ( count ( \$_GET ) != 0 )</b>      ... if ( ci sono parametri ) ...</p>
<p style="text-align: center;"><b>Come il Server Recupera i Parametri Dall'URL (ASPX)</b></p>	<p>In una pagina web dinamica di tipo ASPX (in linguaggio C#) si utilizza la <b>proprietà QueryString</b> dell'<b>oggetto Request</b>, anch'esso sempre disponibile:</p> <p><b>Request.QueryString [ "&lt;nome-parametro&gt;" ]</b> ...restituisce il <b>valore</b> del parametro</p> <p>Esempio:  <b>int ValoreDelMese = (int) Request.QueryString [ "mese" ];</b>  <b>int ValoreDelGiorno = (int) Request.QueryString [ "giorno" ];</b></p> <p>Per conoscere il <b>numero di parametri disponibili</b>, si usa la proprietà <b>Count</b></p> <p>Esempio:  <b>if ( Request.QueryString.Count != 0 )</b>      ... if ( ci sono parametri ) ...</p>

Come il  
Browser  
utilizza il  
Metodo GET

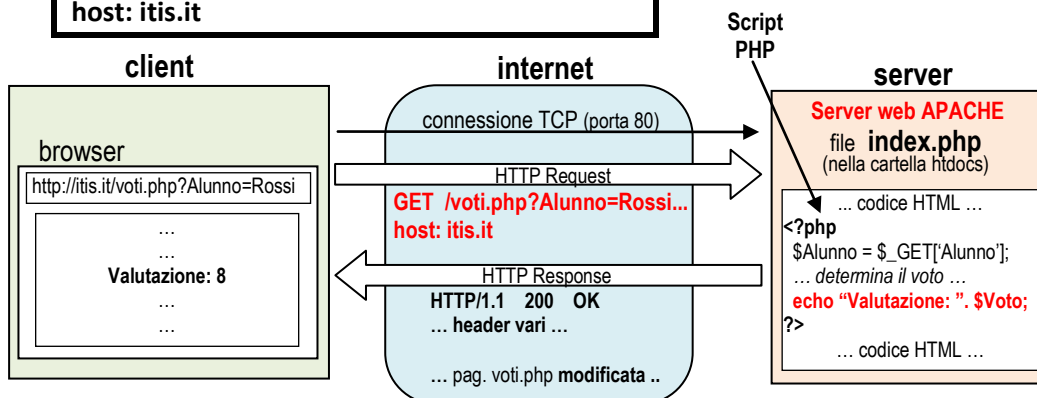
Se **digitiamo un URL** (o clicchiamo su un link), il **browser** apre la connessione TCP sulla porta 80 e invia automaticamente al server una **Request con metodo GET**:

```
GET /pg.html HTTP/1.1
host: itis.it
```



Se digitiamo un **URL con Parametri** (o facciamo click su un link con un URL con Parametri), il **browser** apre la connessione TCP sulla porta 80 e invia al server una **Request basata su un metodo GET con Parametri nell'URL**:

```
GET /voti.php?Alunno=Rossi HTTP/1.1
host: itis.it
```



Creare una  
Pagina Web  
che Invia  
Dati al server  
(form html)

E' necessario utilizzare il tag **<form>** all'interno del quale si dispongono dei tag **<input>** fra i quali deve essere presente uno speciale pulsante di tipo **submit**.

```
<form method="GET" action="calcola.php">
  Base: <input type="text" name="txtB"> <br />
  Altezza: <input type="text" name="txtH"> <br />
  <input type="submit" value="Area Triangolo">
</form>
```

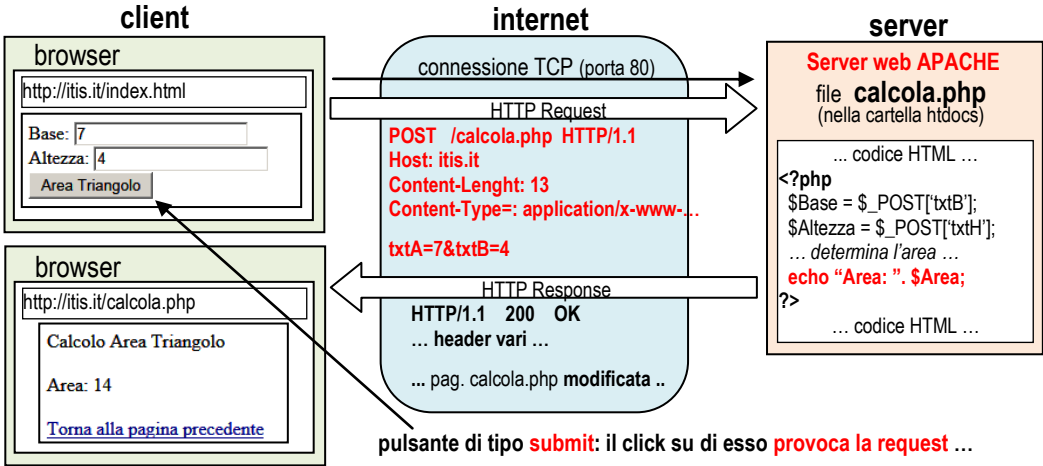
Base:	<input type="text" value="7"/>
Altezza:	<input type="text" value="4"/>
<input type="button" value="Area Triangolo"/>	

L'utente, con questa pagina, può **inserire i dati nelle caselle** (tipo **text**) di nome **txtA** e **txtB**. Al click sul pulsante di tipo **submit**, il browser **invia al server una Request HTTP**:

- basata sul **metodo** specificato nell'attributo **method** (**GET** o **POST**)
- che richiede la **risorsa** specificata nell'attributo **action** (**calcola.php**)
- che riporta nei **parametri** i dati inseriti in **tutti i tag <input>** dotati di nome, secondo il formato **application/x-www-form-urlencoded**, già visto in precedenza (**<nome>=<valore>& ...**)

```
GET /calcola.php?txtB=7&txtH=4 HTTP/1.1
host: itis.it
```

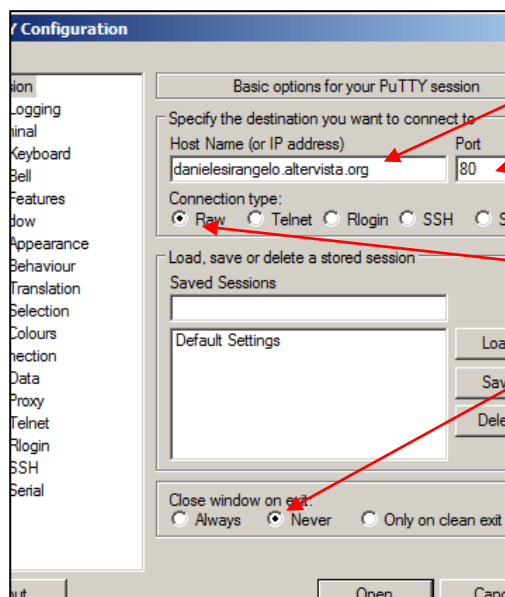
Lo script in **Calcola.php**, *recupera i parametri*, calcola l'area e invia la pagina con il risultato.

<p>Come è Strutturata una Request con Metodo POST</p>	<p>Il <b>metodo POST</b> consente l'invio dei dati all'interno del <b>Corpo</b> della Request (invece di specificarli nell'URL).</p> <p><b>POST</b> &lt;risorsa&gt; <b>HTTP/1.1</b>  <b>Host:</b> &lt;dominio&gt;  <b>Content-Length:</b> &lt;numero di byte presenti nel corpo&gt;  <b>Content-Type:</b> &lt;tipo di contenuto (MIME)&gt;  ... altri possibili Headers ...  &lt;riga vuota&gt;  &lt;dati&gt;</p> <p>Oltre all'header <b>host</b>, che specifica il <b>dominio</b> (cioè il "sito web") in cui cercare la risorsa, sono obbligatori anche:</p> <p>&lt;Content-Length&gt;  Notifica la <i>lunghezza dei dati</i>, ossia il <b>numero di byte presenti nel Corpo</b> della Request.</p> <p>&lt;Content-Type&gt;  Specifica la <b>natura e il formato dei dati</b> trasmessi: lo standard MIME ha stabilito una serie di tipologie, associando a ciascuna di esse una specifica dicitura: <b>text/xml</b> (dati in xml), <b>image/jpeg</b> (immagine), <b>application/x-www-form-urlencoded</b> (formato parametri per form e/o URL), ecc.</p>
<p>Come il Server Web Recupera i Parametri dal Corpo di una Request con Metodo POST</p>	<p>Anche per il Post, la risorsa richiesta è una <b>pagina web dinamica</b> che prevede l'<b>esecuzione di codice sul server</b>. In esso, per "recuperare" il valore dei parametri, si utilizza:</p> <ul style="list-style-type: none"> <li>- In uno script <b>PHP</b> - la variabile super-globale <b>\$_POST</b> (analoga a \$_GET)</li> <li>- In <b>C#</b> (pagina <b>ASPX</b>) - la proprietà <b>Request.Form</b> (analoga a Request.QueryString)</li> </ul>
<p>Come il Browser utilizza il Metodo POST</p>	<p>Se, nell'esempio precedente (<i>Area Triangolo</i>), si specifica <b>method="POST"</b>, allora il browser inoltra la Request basandola sul <i>metodo POST</i> e inserisce i dati nel <i>Corpo della Request</i>, sempre usando il formato <b>application/x-www-form-urlencoded</b>:</p> <div data-bbox="454 1279 1198 1509" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre>POST /calcola.php HTTP/1.1 host: itis.it content-Lenght: 13 content-Type=: application/x-www-form-urlencoded  txtB=7&amp;txtH=4</pre> </div>  <p>The diagram shows the interaction between a client browser, the internet, and a server. The client browser (http://itis.it/index.html) has a form with 'Base: 7' and 'Altezza: 4' and a button 'Area Triangolo'. A 'pulsante di tipo submit' triggers the request. The internet shows the 'connezzione TCP (porta 80)' and the 'HTTP Request' with the following details: <b>POST /calcola.php HTTP/1.1</b>, <b>Host: itis.it</b>, <b>Content-Lenght: 13</b>, <b>Content-Type=: application/x-www-...</b>, and <b>txtA=7&amp;txtB=4</b>. The server (Server web APACHE, file <b>calcola.php</b>) receives the request and returns an 'HTTP Response' with <b>HTTP/1.1 200 OK</b>, '... header vari ...', and '... pag. calcola.php modificata ..'. The client browser then displays 'Calcolo Area Triangolo' and 'Area: 14' with a link 'Torna alla pagina precedente'.</p>

### Confronto fra GET e POST nell'Invio di Dati al Server Web

- L'invio di parametri nell'URL (**metodo GET**) soffre di alcune **limitazioni** quali la *massima lunghezza dell'URL in alcuni browser* e la *necessità di codificare alcuni caratteri*.
- In compenso, essendo i dati nell'URL, essi sono "registrati" nell'URL e rimangono in caso di **memorizzazione dell'URL nei preferiti e nella cronologia** oppure nel caso di un loro utilizzo all'interno di un **collegamento ipertestuale (link)**.
- L'invio di dati nel Corpo (**metodo POST**) non ha limitazioni né nella lunghezza, né nella tipologia dei dati, che possono essere **parametri**, ma anche **flussi binari** (immagini) e, in generale, una **qualsiasi sequenza di byte**.

Il **software free Putty** consente di **aprire una connessione TCP** con un Server, **inviare manualmente dei dati** e **ricevere le risposte** del server stesso. E' utile, nelle esercitazioni, per inviare manualmente delle Request a un Server Web e osservarne le Response.



Dominio del Server Web  
(o suo Indirizzo IP)

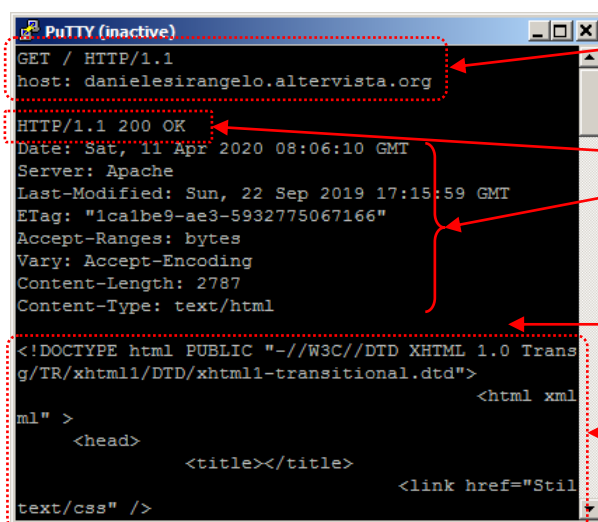
Porta di Rete  
(80 per i Server Web)

Connessione con  
Protocollo TCP

Non Chiudere la  
Finestra dei Dati

### Invio "manuale" di una Request ad un Server Web

per esercitazione  
con software Putty



Digitare manualmente (o incollare)  
il testo della **REQUEST** (metodo GET)

Visionare la **RESPONSE** del Server Web:

Riga di Stato (versione, codice e descrizione stato)

Zona Header (informazioni varie)

Riga Vuota

Corpo (home page del sito, in linguaggio HTML)

N.B.: è preferibile **scrivere preventivamente il testo della Request con il Blocco Note** in modo da effettuare un rapido **copia-incolla** (l'incolla si realizza con il tasto destro) nella finestra dati di Putty: i Server Web, infatti, hanno un **tempo di attesa (Timeout)** minimo e non si ha il tempo di "digitare manualmente" il testo senza incorrere nell'interruzione della connessione da parte del Server stesso.