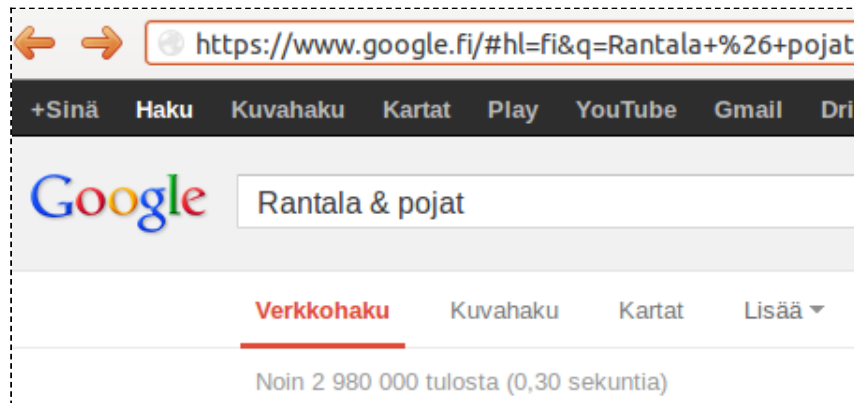


URLin rakenne ja koodaus

Web-ohjelmoija tarvitsee tietoa **URL**in rakenteesta ja epäsallittujen merkkien koodaamisesta, vaikka monin osin selain käsittelee tällaiset asiat automaattisesti. Kuvan mukaisella Google-haulla `Rantala & pojat` huomataan esim. osoiteriviltä, että hakumerkkijono on koodautunut muotoon `Rantala+%26+pojat` eli `&`-merkki on korvattu merkkijonolla `%26` ja välilyönnit `+`-merkeillä.



URL, URI ja URN - mikä ero niillä on?

- URL = Uniform Resource Locator
- URI = Uniform Resource Identifier
- URN = Uniform Resource Name



Jokainen URL on URI, mutta jokainen URI ei ole URL

- URI sisältää sekä URL- että URN-viittaukset
- URN: Yksikäsitteinen resurssin nimi, kuten kirjan ISBN-tunnus. Yksilöi kirjan, mutta ei kerro miten se on saatavilla
- URL: Yksikäsitteinen resurssin paikannin. Kertoo myös miten (protokolla) ja mistä resurssi on saatavilla. Kuvan mukaan dokumentti `ari.html` on saatavilla `http`-protokollaa käyttäen `netisto.fi`-palvelimen hakupolun juuresta `/`. Protokolla voi olla mm. `ftp://`, `sftp://`, `file://` jne.

Yhteenveto: Kaikki URI-viittauksia koskeva dokumentaatio koskee URL-viittauksia ja tässä esityksessä keskitytään HTTP-URL-viittauksiin!

URL-syntaksi

URL:n yleinen syntaksi on:

```
<protokolla>:<protokollaan-liittyvä-osa>
```

Protokollaan liittyvä osa on yleisesti muotoa:

```
//<kayttaja>:<salasana>@<palvelin>:<portti>/<url-polku>
```

HTTP-URL

Tarkastellaan erityisesti HTTP-URLia, jolloin protokollana on http. HTTP-URL:ssa yleisen muodon osa <kayttaja>:<salasana>@ ei ole sallittu ja <url-polku> jakaantuu seuraaviin valinnaisiin osiin:

```
<hakupolku>  
<kyselyosa>
```

Tämän yleinen muoto on

```
http://<palvelin>:<portti>/<hakupolku>?<kyselyosa>
```

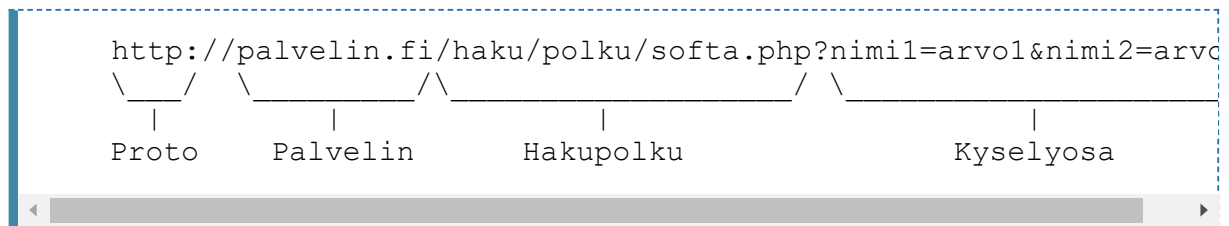
Hakupolku ja kyselyosa (query string) erotetaan toisistaan kysymysmerkillä. Kyselyosuus sisältää nimi=arvo-pareja eroteltuna &-merkillä toisistaan. Osa

```
:<portti>
```

on valinnainen, jos käytetään HTTP-protokollan käyttämää oletusporttia 80.

HTTP-URL pilkottuna

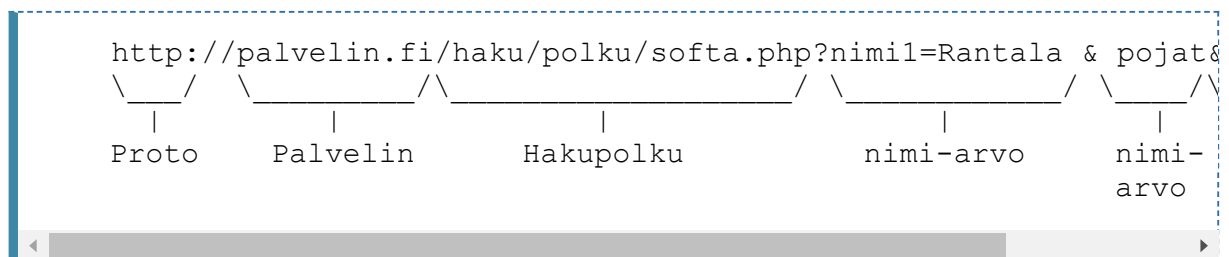
Tavanomainen HTTP-URL voisi näyttää esim. seuraavalta:



Miksi URLin koodaamista

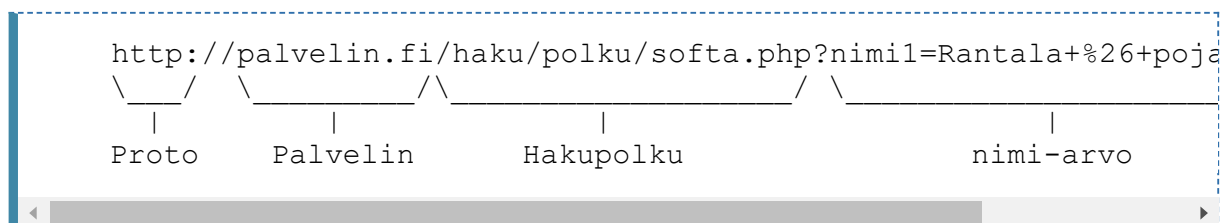
URL sisältää syntaksinsa mukaan useita erikoimerkityksen omaavia merkkejä mm. `?`, `=`, `&`, ... Tällaiset merkit ovat sitten toisaalla URLissa (en)koodattava, jotta niiden käyttö ei rikkoisi halutun URLin merkitystä

Jos edellä esitetyn URLin kyselyosassa parametrin `arvo1` arvona haluttaisiin välittää Rantala & pojat, URL näyttäisi seuraavalta



ja täten `nimi1`-parametrin arvona olisi `Rantala` toisin kuin on haluttu. Lisäksi välilyönnit eivät ole URLissa sallittuja merkkejä, joten URL olisi virheellinen siltäkin osin.

URL-koodaamalla ei-sallitut merkit URL näyttäisi seuraavalta



ja täten `nimi1`-parametrin arvona olisi `Rantala+%26+pojat`, joka olisi oikein sen jälkeen kun erikoismerkit olisi dekoodattu takaisin alkuperäiseen muotoon.

Useimmin selain suorittaa tarvittavan URL-enkoodauksen automaattisesti ja palvelinohjelma vastaavan URL-dekoodauksen. On kuitenkin URLin muodostamisen

tilanteita, joissa selain ei ole apuna. Tällöin URLin koodaamiseen liittyvät asiat tulee hoitaa itse ohjelmoiden.

URLin koodaaminen

Tarkat määritykset vaadittavien merkkien koodaamiseksi perusteluineen löytyvät mm. seuraavista lähteistä

- <https://en.wikipedia.org/wiki/Percent-encoding>
- <https://www.url-encode-decode.com/>
- https://www.w3schools.com/tags/ref_urlencode.asp
- <https://stackoverflow.com/questions/1634271/url-encoding-the-space-character-or-20>

Tässä tyydytään toteamaan, että keskeisimmät URLissa koodattavaksi vaaditut merkit ovat

Merkki	URL-koodaus	Erikoismerkitys
?	%3F	Aloittaa kyselyosan
&	%26	Nimi-arvo-parien erotinmerkki
=	%3D	Yhdistää nimen ja arvon
%	%25	Erikoismerkkien enkoodaus %XX-notaatiolla
+	%2B	esittää välilyöntiä nimi-arvo-pareissa
/	%2F	käytetään hakupolussa
#	%23	viitataan erityiseen kohtaan dokumentissa
välilyönti	%20	jos ilmenee URLissa ennen ?-merkkiä
välilyönti	+	jos ilmenee URLissa jälkeen ?-merkin

URLin koodaaminen PHP:llä

PHP:n tarjoamat funktiot URLin (en)koodaamiseen ja dekoodaamiseen ovat seuraavat

- `urlencode()`
 - Kuten mediatyyppi *application/x-www-form-urlencoded*
 - välilyönnit enkoodataan +-merkeillä.
- `rawurlencode()`
 - Kuten mediatyyppi RFC 2396
 - välilyönnit enkoodataan %20-merkeillä.
- `urldecode()`
- `rawurldecode()`

Seuraavan pienen esimerkkiohjelman avulla voit tutkia PHP:n URLin koodaamiseen tarjoamien funktioiden toimintaa

[urlencode.php](#)

```

1  <form action="urlcode.php" method="get">
2  Syöte:<br>
3  <input type="text" name="indata" value="yks + kaks">
4  <input type="submit">
5  </form>
6
7  <pre>
8  <?php
9  echo ("1:" . $_GET['indata'] . "\n");
10 echo ("2:" . urlencode($_GET['indata']) . "\n");
11 echo ("3:" . rawurlencode($_GET['indata']) . "\n");
12 $indata_urlcoded = urlencode($_GET['indata']);
13 $indata_rawurlcoded = rawurlencode($_GET['indata']);
14 echo ("4:" . urldecode($indata_urlcoded) . "\n");
15 echo ("5:" . rawurldecode($indata_rawurlcoded) . "\n");
16 ?>
17 </pre>

```

Ohessa käyttöesimerkki. Huomaa kokeiluissasi katsoa myös osoiteriville selaimen tekemää URL-koodausta

The screenshot shows a web browser window with the address bar displaying the URL: `netisto.fi/urlcode.php?indata=yks+%2B+kaks`. The page content includes a form with the label "Syöte:" and a text input field containing "yks + kaks". A "Submit" button is next to the input field. Below the form, the output of the PHP script is displayed in a preformatted text block:

```

1:yks + kaks
2:yks+%2B+kaks
3:yks%20%2B%20kaks
4:yks + kaks
5:yks + kaks

```