**ქვიზი 2**

**1. ჰიპერტექსტი - ეს არის:**

ა) დიდი ტექსტი ბ) სტრუქტურირებული ტექსტი კავშირებით

გ) ინტერტექსტი დ) ტექსტი გრაფიკით

**2. ვინ აღწერა პირველად ჰიპერტექსტი:**

ა) დუგლას ენჯელბარტმა ბ) ტედ ნელსონმა

გ) ვენივარ ბუშმა დ) თეოდორ რუზველ­ტმა

**3. რას ამუშავებდა ტედ ნელსონი**

ა) ჰიპერტექსტებში გადასვლების კონცეფციას ბ) მაუსს

გ) მსოფლიო ბიბლიოთეკას დ) ვებ-საიტს

**4. რომელი თარიღი ითვლება WWW დაბადებად:**

ა) 1989 წლის 1 მაისი ბ) 1991 წლის 19 აგვისტო

გ) 1990 წლის 30 ოქტომბერი დ) 1991 წლის 6 აგვისტო

**5.ვინ არის WWW -ს შემქმნელი:**

ა) პოლ ბერენი ბ) ლარი რო­ბერტსის

გ) სერ ტიმოთი ჯონ (ტიმ) ბერნერს-ლი დ) ლეონარდ კლაინროკი

**6. ვინ იყო WWW -ს შექმნის თანამონაწილე**

ა) მარკ ანდრესენი ბ) რობერტ კაიო

გ)დუგლას ენჯელბარტი დ) ბილ გეითსი

**7. სად შეიქმნა WWW:**

ა) კემბრიჯში, ევროპა ბ) მასაჩუსეტში. აშშ

გ) კალიფორნია, სილიკონის ველზე დ) ცერნში, ევროპა

**8. პირველი ვებ-გვერდის მისამართია:**

ა) <http://info.cern.ch/> ბ) <http://inform.carn.ch>

გ) http://www. info.ge დ) http://WWW. cern.ch

**9. რომელ ძირითად პროტოკოლს იყენებს WWW :**

ა) TCP ბ) IP

ბ) FTP დ) HTTP

**10.** **რას წარმოადგენს ბრაუზერი:**

ა) სპეციალურ პროგრამას - ტექნიკურად [„HTTP](http://ka.wikipedia.org/w/index.php?title=HTTP&action=edit&redlink=1) კლიენტს“, რომელიც მომხმარებელს საშუალებას აძლევს[„ HTML დოკუმენტების“](http://ka.wikipedia.org/w/index.php?title=HTML_%E1%83%93%E1%83%9D%E1%83%99%E1%83%A3%E1%83%9B%E1%83%94%E1%83%9C%E1%83%A2%E1%83%98&action=edit&redlink=1) ინფორმაციის ჩვენებას [ვებ-სერვერიდან](http://ka.wikipedia.org/wiki/%E1%83%95%E1%83%94%E1%83%91-%E1%83%A1%E1%83%94%E1%83%A0%E1%83%95%E1%83%94%E1%83%A0%E1%83%98) ან [ფაილური სისტემიდან](http://ka.wikipedia.org/w/index.php?title=%E1%83%A4%E1%83%90%E1%83%98%E1%83%9A%E1%83%A3%E1%83%A0%E1%83%98_%E1%83%A1%E1%83%98%E1%83%A1%E1%83%A2%E1%83%94%E1%83%9B%E1%83%90&action=edit&redlink=1).

ბ) აპარატულ მოწყობილებას, რომელსაც შეუძლია  [HTML დოკუმენტების](http://ka.wikipedia.org/w/index.php?title=HTML_%E1%83%93%E1%83%9D%E1%83%99%E1%83%A3%E1%83%9B%E1%83%94%E1%83%9C%E1%83%A2%E1%83%98&action=edit&redlink=1) ინფორმაციის ჩვენებას [ვებ-სერვერიდან](http://ka.wikipedia.org/wiki/%E1%83%95%E1%83%94%E1%83%91-%E1%83%A1%E1%83%94%E1%83%A0%E1%83%95%E1%83%94%E1%83%A0%E1%83%98) ან [ფაილური სისტემიდან](http://ka.wikipedia.org/w/index.php?title=%E1%83%A4%E1%83%90%E1%83%98%E1%83%9A%E1%83%A3%E1%83%A0%E1%83%98_%E1%83%A1%E1%83%98%E1%83%A1%E1%83%A2%E1%83%94%E1%83%9B%E1%83%90&action=edit&redlink=1).

გ) მათვალიერებელს

დ) მაყურებელს

**11. ბრაუზერის მუშაობის პრინციპი:**

ა) URL მისამართი-კეშმეხსიერება - HOSTSფაილი - DNS- ვებ გვერდი

ბ) URL მისამართი- HOSTSფაილი -კეშმეხსიერება - DNS- ვებ გვერდი

გ) DNS - URL მისამართი-კეშმეხსიერება - HOSTSფაილი - ვებ გვერდი

დ) კეშმეხსიერება - HOSTSფაილი - ვებ გვერდი - DNS - URL მისამართი

**12. HTTP პროტოკოლის მუშაობა**

ა) TCP პროტოლოლით უკავშირდება სერვერს 80-პორტით, აგზავნის HTTP მოთხოვნის სტრიქონს; სერვერიდან ღებულობს HTTP პასუხს.

ბ) TCP პროტოლოლით უკავშირდება სერვერს 220-პორტით, აგზავნის HTTP მოთხოვნის სტრიქონს; სერვერიდან ღებულობს HTTP პასუხს.

გ) აგზავნის HTTP მოთხოვნის სტრიქონს; სერვერიდან ღებულობს HTTP პასუხს. დ) FTP პროტოლოლით უკავშირდება სერვერს 80-პორტით, აგზავნის HTTP მოთხოვნის სტრიქონს; სერვერიდან ღებულობს HTTP პასუხს.

**13**. HTTP მოთხოვნა-პასუხს აქვს შემდეგი სტრუქტურა

ა) სათაური (ინგ. *Headers*) - ახასიათებენ შეტყობინების ტანს, გადაცემის პარა­მეტრებს და სხვა. სასტარტო სტრიქონი (ინგ. *Starting line*) განსაზღვრავს შეტყობინების (მოთ­ხოვნის) ტიპს; სათაური (ინგ. *Headers*) - ახასიათებენ შეტყობინების ტანს, გადაცემის პარა­მეტრებს და სხვა.

ბ) სასტარტო სტრიქონი (ინგ. *Starting line*) განსაზღვრავს შეტყობინების (მოთ­ხოვნის) ტიპს; სათაური (ინგ. *Headers*) - ახასიათებენ შეტყობინების ტანს, გადაცემის პარა­მეტრებს და სხვა. შეტყობინების ტანი (ინგ. *Message Body*) - უშუალოდ შეტყობინების მონაცემე­ბი, რომელიც სათაურისგან აუცილებლად უნდა იყოს დაშორებული ცარიელი სტრიქონით

გ) სასტარტო სტრიქონი (ინგ. *Starting line*) განსაზღვრავს შეტყობინების (მოთ­ხოვნის) ტიპს; შეტყობინების ტანი (ინგ. *Message Body*) - უშუალოდ შეტყობინების მონაცემე­ბი, რომელიც სათაურისგან აუცილებლად უნდა იყოს დაშორებული ცარიელი სტრიქონით,სათაური (ინგ. *Headers*) - ახასიათებენ შეტყობინების ტანს, გადაცემის პარა­მეტრებს და სხვა.

დ) სასტარტო სტრიქონი (ინგ. *Starting line*) განსაზღვრავს შეტყობინების (მოთ­ხოვნის) ტიპს; შეტყობინების ტანი (ინგ. *Message Body*) - უშუალოდ შეტყობინების მონაცემე­ბი, რომელიც სათაურისგან აუცილებლად უნდა იყოს დაშორებული ცარიელი სტრიქონით; სათაური (ინგ. *Headers*) - ახასიათებენ შეტყობინების ტანს, გადაცემის პარა­მეტრებს და სხვა.

14. **ტრადიციული ფორმა**

ა) გამოყენებითი **<სქემა>://<პაროლი><ლოგინი>::@<ჰოსტი>:<პორტი>/<URL-გზა>? <პარამეტრები>#<ღუზა>**

ბ)**სქემა>://<პარამეტრები>#<ღუზა><ლოგინი>:<პაროლი>:@<ჰოსტი>:<პორტი>/<URL-გზა>?**

გ) **<სქემა>://<ლოგინი>:<პაროლი>:@<ჰოსტი>:<პორტი>/<URL-გზა>? <პარამეტრები>#<ღუზა>**

დ) **<სქემა>://<ლოგინი>:<პაროლი>:@<ჰოსტი>:<პორტი>/<URL-გზა>? <პარამეტრები>#<ღუზა>**

**15. რა არის URI**

ა) ან URL ან **URN** (ინგ.***Uniform Resource Name***- რესურსის ერთიანი (უნიფიცირებული) სახელი), ანაც ორივე ერთდროულად

ბ) ან URL

გ) ან **URN**

დ) ან **IPv4**

**16.** **HTTP პროტო­კოლის საფუძველს წარმოადგენს**

ა) ”კლიენტ-კლიენტი” ტექნოლოგია ; ბ) ”კლიენტ-სერვერის” ;

გ) ”კლიენტ-სერვერის” ტექნოლოგია ; დ) ”სერვერ-სერვარ” ტექნოლოგიურად:

17. რა ფორმა გააჩნია მოთხოვნის სასტარტო სტრიქონს

ა) მეთოდი<SP>URI<SP>HTTP/ვერსია<CRLF>

ბ) URI<SP>HTTP/ვერსია ,მეთოდი<SP> HTTP/ვერსია

გ) მეთოდი<SP> CRLF <SP>HTTP/ვერსია<CRLF URI >

დ) მეთოდი<CRLF URI >

18) **რა ფორმა გააჩნია პასუხის სასტარტო სტრიქონს**

ა) HTTP/ვერსია< SP>მდგომარეობის კოდი< SP> განმარტება

ბ) განმარტება< SP>მდგომარეობის კოდი< SP> HTTP/ვერსია

გ) მდგომარეობის კოდი< SP> HTTP/ვერსია< SP> განმარტება

დ)< SP>მდგომარეობის კოდი< SP> HTTP/ვერსია< SP> განმარტება

19) **რომელია სწორი:**

ა) **GET** – გამოიყენება URI მოთხოვნაში მითითებული ინფორმაციის მისაღებად.

**HEAD -** იგივეა რაც GET, მაგრამ სერვერი პასუხისას არ აბრუნებს შეტყობინება-პასუხის ტანს, მხოლოდ პასუხის სათაურს აგზავნის.

**POST -** გამოიყენება რესურსისთვის მომხმარებლის მონაცემთა გადასაგზავნათ. მაგალითად, ბლოგებში, სტუმარს შეუძლია ჩაწეროს თავისი კომენტარი, რის შემდეგაც ისინი გადაიგზავნებიან სერვერზე POST მეთოდით და ისინი დაიდება ვებ-გვერდზე.

**PUT -** გამოიყენება შეტყობინების ტანის ჩასატვირთად მითითებულ URI-ში. თუ კი მითითებულ URI-ში არ არის რესურსი, იგი ქმნის ამ რესურს და აბრუნებს პასუხში პარამეტრს 201(Created - შექმნილის).

ბ) **GET-**გამოიყენება რესურსისთვის მომხმარებლის მონაცემთა გადასაგზავნათ. მაგალითად, ბლოგებში, სტუმარს შეუძლია ჩაწეროს თავისი კომენტარი, რის შემდეგაც ისინი გადაიგზავნებიან სერვერზე POST მეთოდით და ისინი დაიდება ვებ-გვერდზე.

**POST**– გამოიყენება URI მოთხოვნაში მითითებული ინფორმაციის მისაღებად.

**PUT -** გამოიყენება შეტყობინების ტანის ჩასატვირთად მითითებულ URI-ში. თუ კი მითითებულ URI-ში არ არის რესურსი, იგი ქმნის ამ რესურს და აბრუნებს პასუხში პარამეტრს 201(Created - შექმნილის).

**HEAD -** იგივეა რაც GET, მაგრამ სერვერი პასუხისას არ აბრუნებს შეტყობინება-პასუხის ტანს, მხოლოდ პასუხის სათაურს აგზავნის.

**გ)** GET - იგივეა რაცHEAD, მაგრამ სერვერი პასუხისას არ აბრუნებს შეტყობინება-პასუხის ტანს, მხოლოდ პასუხის სათაურს აგზავნის.

**POST**– გამოიყენება URI მოთხოვნაში მითითებული ინფორმაციის მისაღებად.

**PUT -** გამოიყენება შეტყობინების ტანის ჩასატვირთად მითითებულ URI-ში. თუ კი მითითებულ URI-ში არ არის რესურსი, იგი ქმნის ამ რესურს და აბრუნებს პასუხში პარამეტრს 201(Created - შექმნილის).

**HEAD -** იგივეა რაც GET, მაგრამ სერვერი პასუხისას არ აბრუნებს შეტყობინება-პასუხის ტანს, მხოლოდ პასუხის სათაურს აგზავნის.

დ) **GET** – გამოიყენება URI მოთხოვნაში მითითებული ინფორმაციის მისაღებად.

**HEAD -** იგივეა რაც GET, მაგრამ სერვერი პასუხისას არ აბრუნებს შეტყობინება-პასუხის ტანს, მხოლოდ პასუხის სათაურს აგზავნის.

**POST -** გამოიყენება რესურსისთვის მომხმარებლის მონაცემთა გადასაგზავნათ. მაგალითად, ბლოგებში, სტუმარს შეუძლია ჩაწეროს თავისი კომენტარი, რის შემდეგაც ისინი გადაიგზავნებიან სერვერზე POST მეთოდით და ისინი დაიდება ვებ-გვერდზე.

**PUT -** გამოიყენება შეტყობინების ტანის ჩასატვირთად მითითებულ URI-ში. თუ კი მითითებულ URI-ში არ არის რესურსი, იგი ქმნის ამ რესურს და აბრუნებს URI პასუხში პარამეტრს 201(Created - შექმნილის).

20. რა გაფართოვება აქვს HTML ფაილს

ა) .html

ბ) .html ან .htm

გ) .htm

დ) .txt