

# ОСНОВЫ web

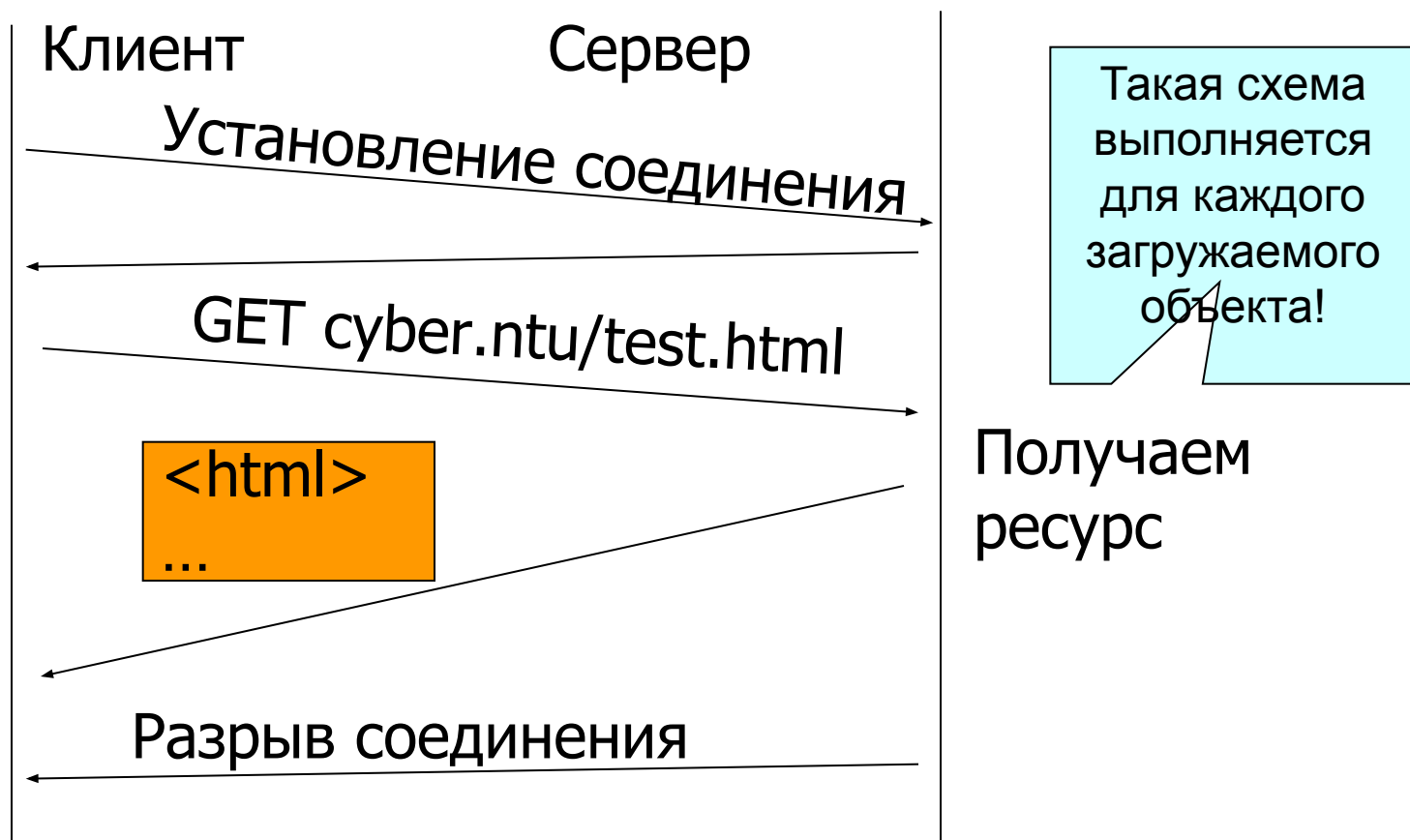


## Протокол HTTP

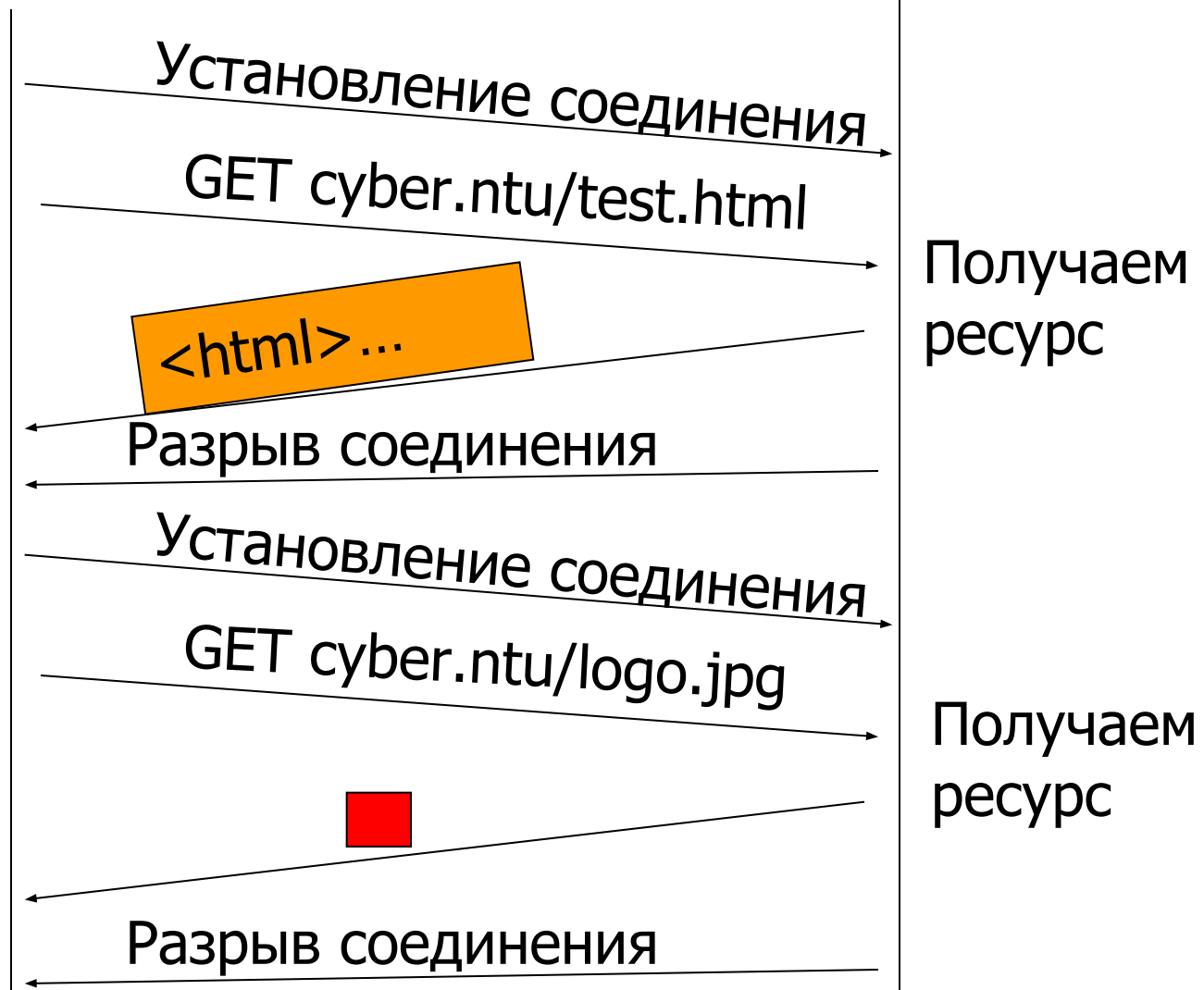
Viacheslav Kolbasin, <EPAM>

# Протокол HTTP

- Является протоколом уровня приложения над транспортным протоколом TCP
- Построен по запрос-ответной схеме



# Протокол HTTP (2)

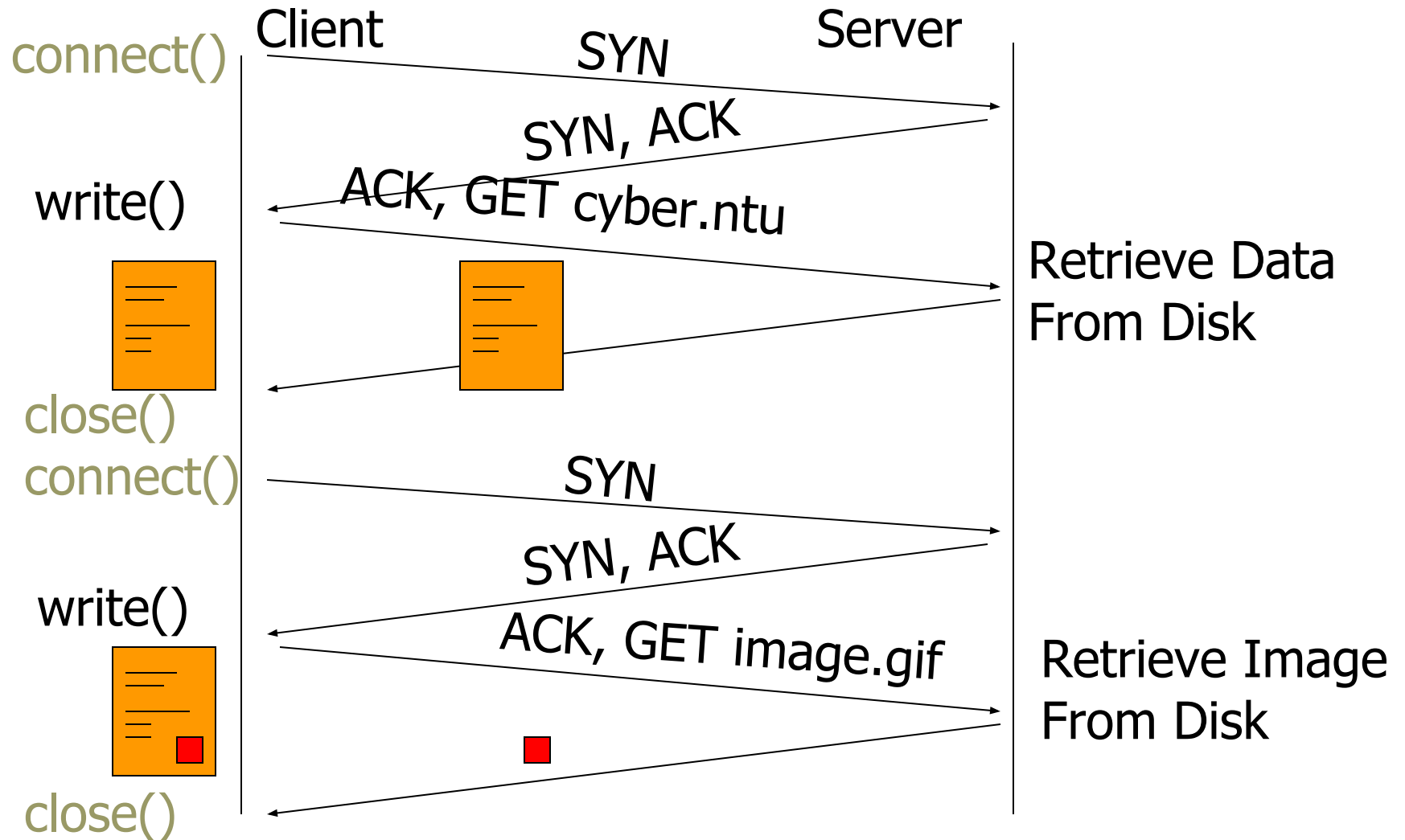


# Особенности протокола

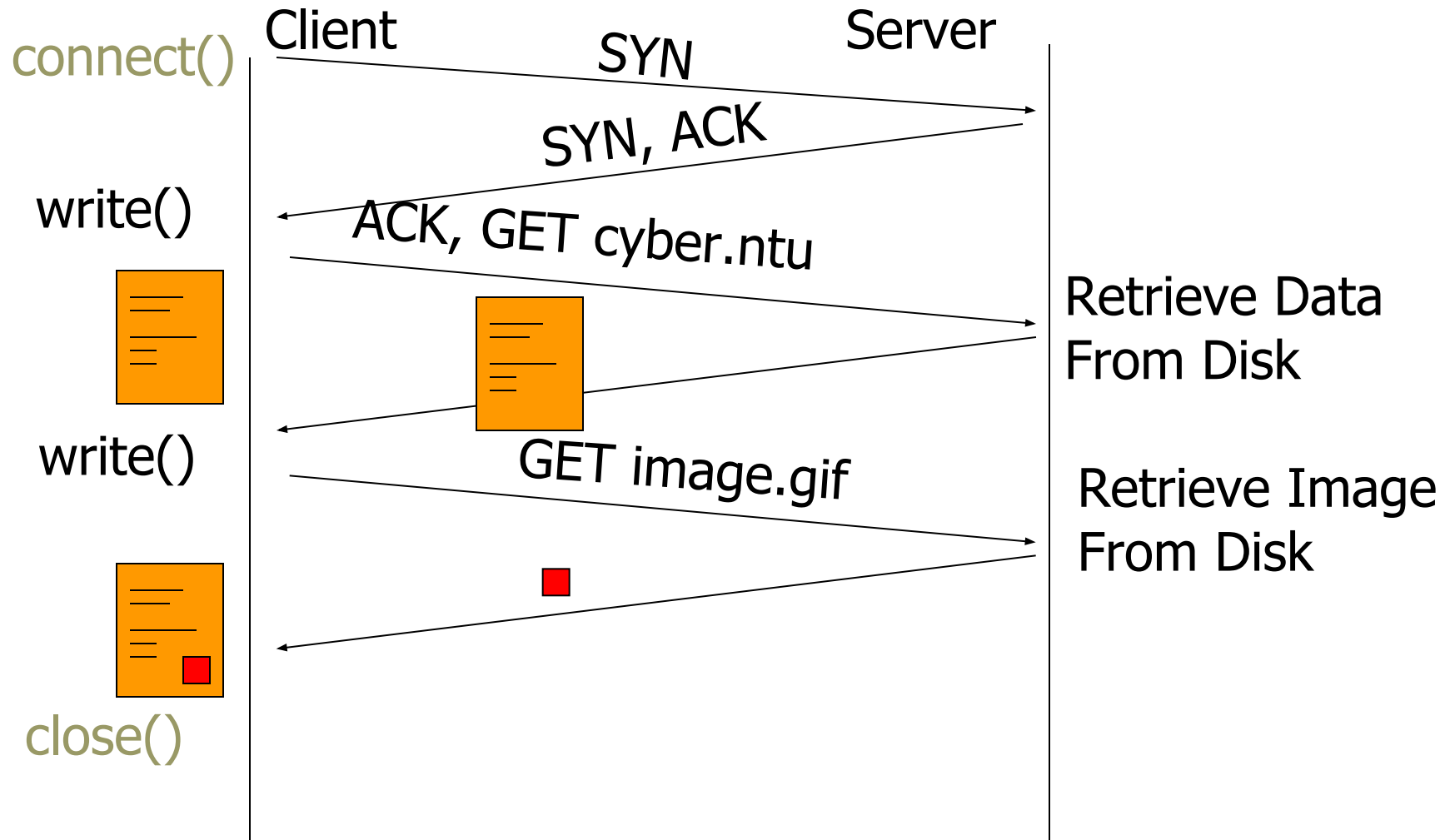
---

- Протокол не имеет состояния
  - То есть нет *штатных* средств для соединения подряд идущих запросов в единое целое
- Существенный перерасход ресурсов при большом количестве объектов на странице
  - Надо открывать/закрывать соединения для каждого объекта
- Версии протокола
  - 0.9 – первая версия протокола, реально применявшаяся в сети Интернет
    - Поддерживала только чтение ресурсов
  - 1.0 – появилась поддержка других операция, кроме чтения
    - Но осталось требования отдельного соединения для каждого объекта
  - 1.1 – современная версия
    - Появилась возможность считывать объекты в рамках одного открытого соединения

# HTTP 1.0



# HTTP 1.1



# Формат сообщений

---

- Сообщения протокола HTTP имеют следующую структуру
  - *Строка-Статуса* **CRLF**
  - *Поля заголовков* **CRLF**
  - *CRLF*
  - *[Содержание-Запроса]*
  
- Для запроса строка статуса состоит из
  - *Метод \_ URI-Запроса \_ Версия-HTTP* **CRLF**
- Для ответа:
  - *Версия-HTTP \_ Код-статуса \_ Фраза-Объяснение*
  
- Заголовки частично дублируются, то есть могут применяться как в запросе, так и в ответе

# HTTP запрос (1)

---

- Строка статуса в запросе выглядит так:
  - `Метод _ URI-Запроса _ Версия-HTTP CRLF`
  
- **Метод** определяет, что именно будет сделано
- Основные методы
  - GET - читать ресурс
  - POST - отправить данные и читать ресурс
  - HEAD - считать только заголовки для ресурса
  - PUT - разместить ресурс
  - DELETE - и удалить ресурс
  
- **URI-Запроса** определяет путь к ресурсу внутри сервера
- **Версия-HTTP** задает используемую версию протокола
  - Скорее всего сейчас везде будет равняться «HTTP/1.1»



# Заголовки запроса

---

- Заголовки имеют следующий общий вид:
  - Название : значение
  
- ***If-Modified-Since*** : дата и время
  - Заголовок реализует «Условный GET»
  - В ответ на условный GET, тело запрашиваемого ресурса передается только, если он изменялся после даты, указанной в заголовке "***If-Modified-Since***".
    - Если значение времени в поле заголовка некорректно, ответ будет идентичен ответу на обычный запрос GET.
    - Если после указанной даты ресурс изменялся, ответ будет также идентичен ответу на обычный запрос GET.
    - Если ресурс не изменялся после указанной даты, сервер вернет код статуса "304 Not Modified".

# Заголовки запроса (2)

---

## ▣ **Accept** : MIME-тип

- Список воспринимаемых клиентом форматов тела ответа
- Например:
  - ▣ `Accept: text/html; text/plain; image/jpeg`

## ▣ **From** : e-mail пользователя

## ▣ **Host** : имя сервера

- Используется для задания символьного имени сервера, к которому идет обращение
- Полезно, есть несколько виртуальных хостов на одном IP-адресе

## ▣ **Referer** : URL-адрес

- Содержит адрес страницы, откуда был выполнен переход на запрашиваемый ресурс
- Часто *использовался* для ограничения доступа к некоторым ресурсам только с определенной веб-страницы

# Заголовки запроса (3)

---

- ▣ **User-Agent** : название браузера
  - Поле содержит информацию о пользовательском агенте, пославшем запрос.
  - Например:
    - ▣ User-Agent: CERN-LineMode/2.15 libwww/2.17b3
  
- ▣ **Connection** : close | Keep-alive
  - Определяет, должен ли сервер закрывать соединение после отправки тела ответа
  
- ▣ **Content-Type** : MIME-тип
  - Задаёт тип тела ответа и **кодировку** !
  - Например:
    - ▣ Content-type: text/html; charset=windows-1251
  
- ▣ **Content-Length** : размер тела ответа в байтах

# Заголовки запроса (3)

---

- ▣ **User-Agent** : название браузера
  - Поле содержит информацию о пользовательском агенте, пославшем запрос.
  - Например:
    - ▣ User-Agent: CERN-LineMode/2.15 libwww/2.17b3
  
- ▣ **Connection** : close | Keep-alive
  - Определяет, должен ли сервер закрывать соединение после отправки тела ответа
  
- ▣ **Content-Type** : MIME-тип
  - Задаёт тип тела ответа и **кодировку** !
  - Например:
    - ▣ Content-type: text/html; charset=windows-1251
  
- ▣ **Content-Length** : размер тела ответа в байтах

# HTTP ответ (1)

---

- Строка статуса в ответе выглядит так:
  - Версия-HTTP \_ Код-статуса \_ Фраза-Объяснение
  
- **Версия-HTTP** задает используемую версию протокола
  - Скорее всего сейчас везде будет равняться «HTTP/1.1»
  
- **Код статуса** определяет код результата выполнения запроса
  - Предназначен для машины
  
- Фраза-объяснение также определяет результат выполнения запроса, но
  - предназначена для человека

# Коды статуса

---

- ▣ **1xx**: Информационный - Не используется, зарезервирован
- ▣ **2xx**: Успех - Запрос был полностью получен, понят, и принят к обработке.
- ▣ **3xx**: Перенаправление - Клиенту следует предпринять дальнейшие действия для успешного выполнения запроса.
- ▣ **4xx**: Ошибка клиента - Запрос, содержащий неправильные синтаксические конструкции, не может быть успешно выполнен.
  - ошибка была допущена со стороны клиента.
- ▣ **5xx**: Ошибка сервера - Сервер не смог дать ответ на корректно поставленный запрос.
  - сервер допустил ошибку

# Status Codes

---

- 200 OK
- 201 created
- 202 accepted
- 204 no content
- 301 moved perm.
- 302 moved temp
- 304 not modified
- 400 bad request
- 401 unauthorized
- 403 forbidden
- 404 not found
- 500 int. server error
- 501 not impl.
- 502 bad gateway
- 503 svc not avail

# Заголовки ответа (1)

---

- ▣ **Last-Modified** : время и дата
  - Поле заголовка содержит дату и время, в которое, по мнению отправляющей стороны, ресурс был последний раз модифицирован
  - Например:
    - ▣ Last-Modified: Tue, 15 Nov 2004 12:45:26 GMT
  
- ▣ **Location** : URL-адрес
  - Для значений кода статуса **3xx**, определяет адрес, куда был перемещен ресурс
  - Например:
    - ▣ Location: <http://cyber.ntu/moved-sites/index.html>
  
- ▣ **Server** : название сервера
  - Содержит информацию о сервере



# Заголовки ответа (2)

---

- ▣ **Connection** : close | Keep-alive
  - Определяет, должен ли сервер закрывать соединение после отправки тела ответа
  
- ▣ **Content-Type** : MIME-тип
  - Задаёт тип тела ответа и **кодировку** !
  - Например:
    - ▣ Content-type: text/html; charset=windows-1251
  
- ▣ **Content-Length** : размер тела ответа в байтах
  
- ▣ **Cache-control** : команда
  - Задаёт директивы управления кешированием
  - Основной вариант использования:
    - ▣ Cache-Control: no-cache; no-store

# Заголовки ответа (3)

---

## ▣ **Cache-control** : команда

- Задаёт директивы управления кешированием
- Директивы управления кешированием:
  - ▣ public - документ кешируется во всех доступных кэшах
  - ▣ private - кешируется браузером, не кешируется проху-сервером
  - ▣ no-cache - запрет на кеширование (браузер и проху-сервер документ не кешируют)
  - ▣ no-store - может быть кеширован, но не сохраняется в архиве
  - ▣ max-age=*time*, must-revalidate - указывает браузеру сколько секунд хранить документ в кэше
  - ▣ max-age=*time*, proxy-revalidate - указывает сколько секунд хранить документ в кэше проху-серверу