

# HTTP

# 学習内容

---

- 1 HTTPの基礎知識
- 2 Webアクセスとメッセージの構造
- 3 高速化を実現したHTTP/2の仕組み
- 4 次世代プロトコルHTTP/3の革新

0

1

## 1.0 HTTPの基礎知識

# HTTPとは？Webのデータを運ぶプロトコル

---

HTTPはWebサーバとブラウザ間の通信を定義するプロトコル

**HTTP**（HyperText Transfer Protocol）は、Web通信の基本プロトコルである

WebサーバとWebブラウザの間でデータをやり取りするために利用される

**HTMLファイル**、画像、音声、動画などのあらゆるデータを送受信できる

技術的な土台は**TCP**の上で動作しており、代表的なポート番号は**80**（http）や**443**（https）

# インターネット資源の位置情報：URLの構造

URLはインターネット上の資源を一意に特定する

構成要素	説明	例
スキーム	利用するプロトコルを示す（http, httpsなど）	https
ホスト名	アクセス先のサーバ名	www.infraexpert.com
ポート番号	接続先のポート番号（省略可能）	443（httpsのデフォルト）
パス	ファイル名やディレクトリの指定	/index.html

02

## 2.0 Webアクセスとメッセージの構造

# Webアクセスの仕組み：リクエストとレスポンス

---

ブラウザとサーバの間で交わされる一連の流れ

STEP 1

ユーザーがブラウザにURLを入力



STEP 2

ブラウザが**HTTPリクエスト**を送信



STEP 3

サーバがリクエストを解析・処理

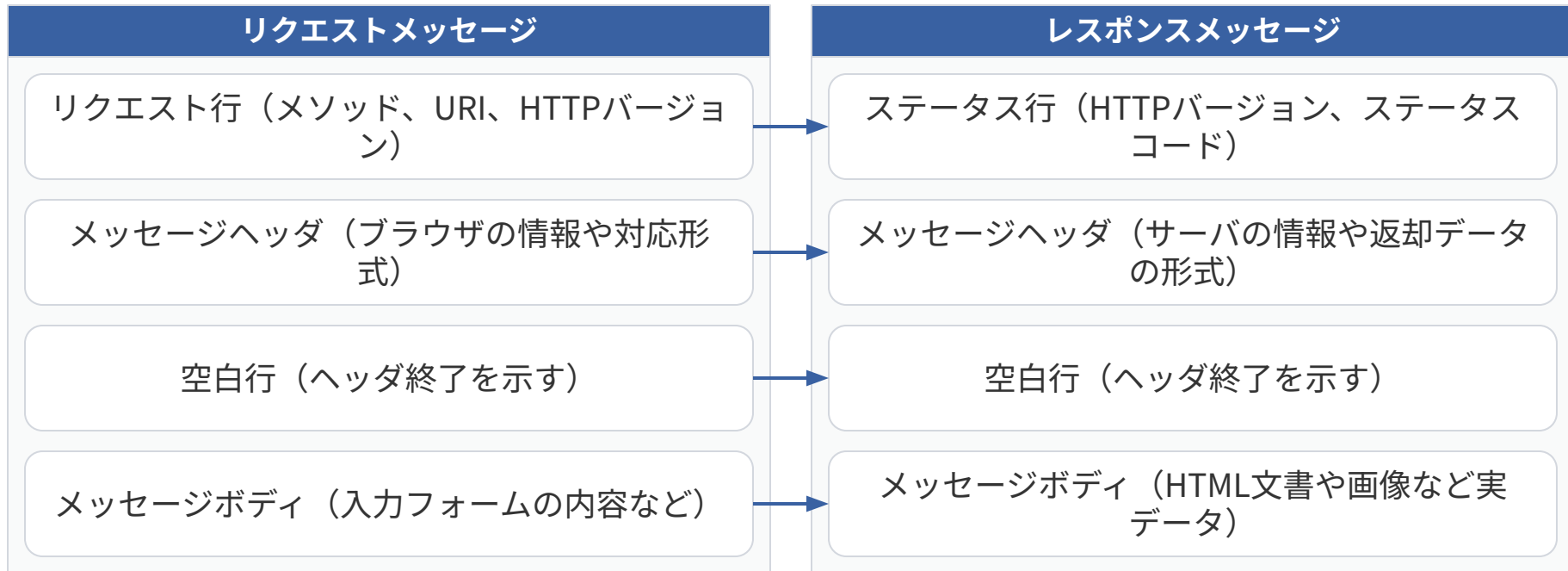


STEP 4

サーバが**HTTPレスポンス**を返送

# HTTPメッセージの構成要素

## リクエストとレスポンスのメッセージ構造





# 代表的なHTTPメソッド

---

Web上で実現したい操作を定義する

**GET**

データの取得

**POST**

データを送信

**HEAD**

ヘッダ情報のみ取得

**PUT**

ファイルをアップロード

**DELETE**

データを削除

# HTTPステータスコードの分類

---

リクエストの結果を示す3桁の数字コード

100番台

**情報**

100 Continue

200番台

**成功**

200 OK

300番台

**リダイレクト**

301 Moved

400番台

**クライアント  
エラー**

404 Not Found

500番台

**サーバエラー**

503 Service  
Unavailable

# 03

## 3.0 高速化を実現したHTTP/2の仕組み

# HTTPバージョンの変遷（TCPベース）

## HTTP/1.1からHTTP/2への進化

### HTTP/1.1の特徴

リクエストごとに**TCP接続を切断**

順次処理のため『**HOLブロッキング**』が発生しやすい

テキストベースでヘッダが大きい

クライアントがリクエストしないとデータは来ない

### HTTP/2の特徴

1つのTCP接続で**多重化**通信を実現

リクエスト・レスポンスの同時処理が可能

HPACKによる**ヘッダ圧縮**

**サーバプッシュ**による先送りが可能

# HTTP/2の主要な仕組み：ストリームとフレーム

通信を効率化するための新しい概念

## ストリーム

1つのTCP接続上で、複数のリクエストとレスポンスを同時にやり取りする論理的な流れ

## フレーム

通信の最小単位。テキストではなくバイナリ形式でデータをやり取りし、ヘッダやデータを識別

# HTTP/2のその他のメリット

---

高速化・効率化に貢献する機能

- 1 **サーバプッシュ:** クライアントがリクエストする前に、サーバから関連データを先に送信する
- 2 **優先度制御:** ストリームごとに優先度を指定し、重要なデータを先に送る
- 3 **ヘッダ圧縮 (HPACK) :** 繰り返し送られるヘッダ情報を圧縮し、通信量を削減する
- 4 **TLS利用のルール:** セキュリティのためにTLS 1.2以上などの要件がある

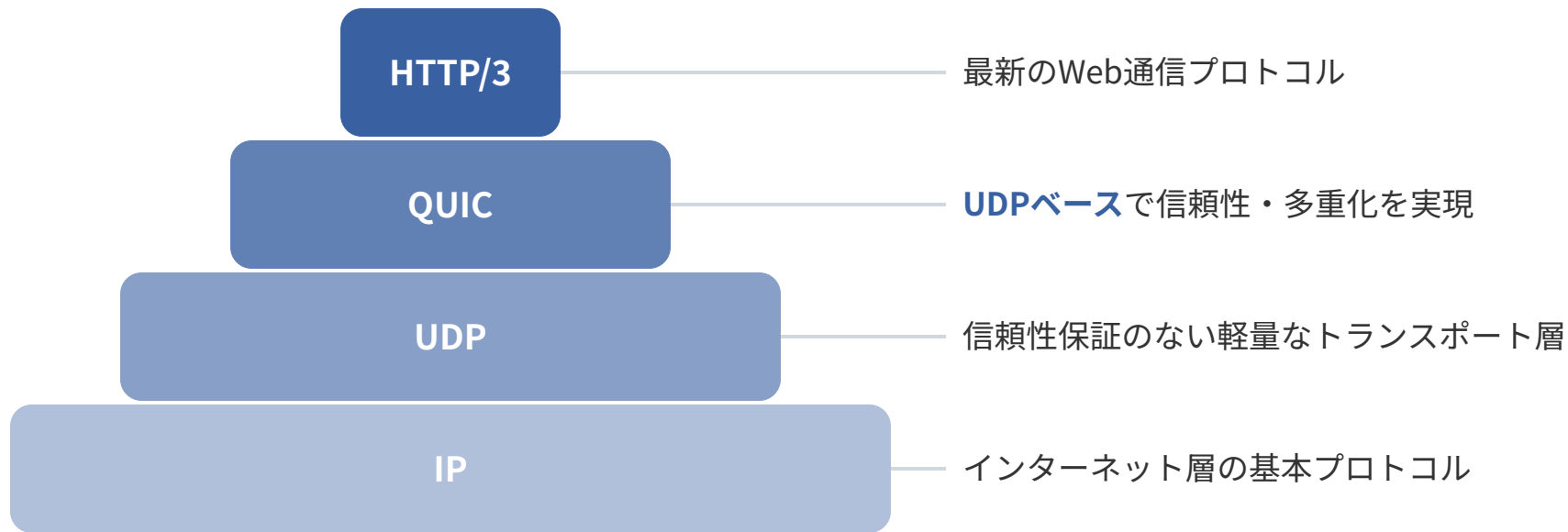
04

# 4.0 次世代プロトコルHTTP/3の革新

# HTTP/3を支える新しいプロトコルスタック

---

TCPからUDPとQUICへの基盤変更





# HTTP/2とHTTP/3の比較

基盤プロトコルと高速化のメカニズム

## HTTP/2 (TCPベース)

HOLブロッキングが残る

2段階ハンドシェイクが必要

IP/ポート変更で切断

TCP

ブロッキング

接続確立

接続維持

基盤プロトコル

## HTTP/3 (QUIC/UDPベース)

HOLブロッキングを解消

高速接続開始 (0-RTTも可能)

コネクションマイグレーション  
で維持

UDP / QUIC

# QUICによる接続高速化の仕組み

---

## ハンドシェイクと遅延の短縮

- 1 QUICがTCPハンドシェイクを省略し、**TLS 1.3**のハンドシェイクから通信を開始
- 2 初回接続時も**1-RTT**（ラウンドトリップタイム）でデータ送信が可能
- 3 回目以降は**0-RTT再開機能**により即座にデータ送信
- 4 **HOLブロッキング**を回避し、遅延の影響を最小化