

A faint, light blue world map is visible in the background of the slide, centered behind the text.

# INTERNET ACADEMY

Institute of Web Design & Software Services

## Git補足

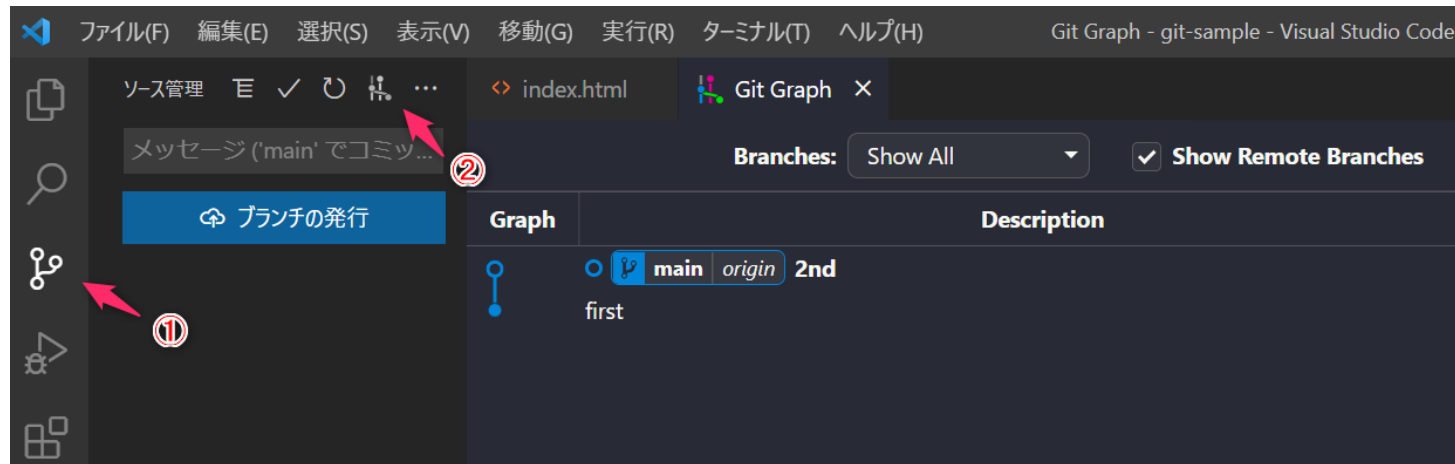
インターネット・アカデミー

A faint, light blue world map is visible in the background of the slide, centered behind the title text.

# VSCodeの補足

# VSCode拡張機能の例

Git Graph コミット履歴を見やすく表示



VSCodeで表示->コマンドパレットでgitlogと入力でもok

(他にもGitLens, GitHistoryなどの拡張機能がある)

A faint, light blue world map is visible in the background of the slide, centered behind the text.

# Git/GitHub

## 初期設定

# Git初期設定 1

Git Bashで実行

```
git config --global user.name 'xxx'
```

```
git config --global user.email 'xxx'
```

## Git初期設定 2 SSHキー生成

Git Bashで実行

```
mkdir ~/.ssh
```

```
cd ~/.ssh
```

```
ssh-keygen -t rsa -b 4096 -C 'メールアドレス'  
-f id_rsa
```

パスフレーズを2回入力すると

公開鍵と秘密鍵を生成できる

## Git初期設定 2 SSHキー生成

C:¥User¥ユーザー名¥.ssh

フォルダ内に

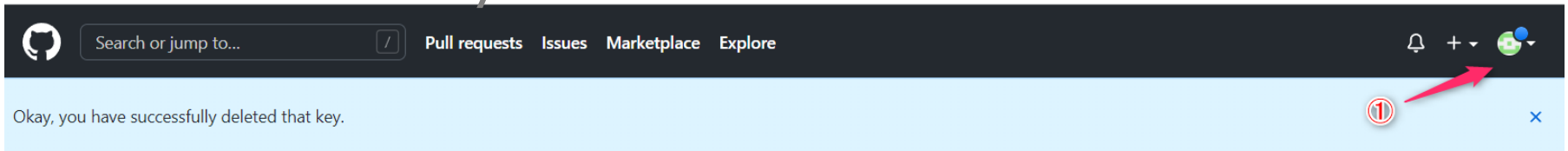
id\_rsaとid\_rsa.pubが

生成される

# GitHubにログイン

GitHubのSSH and GPG keys から

New SSH key をクリック



**AkihiroAoki8**

Your personal account

Public profile

Account

Appearance

Accessibility

Notifications

Access

Billing and plans

Emails

Password and authentication

**SSH and GPG keys**

Organizations



## SSH keys

There are no SSH keys associated with your account.

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).



New SSH key

[Go to your personal profile](#)

## GPG keys

There are no GPG keys associated with your account.

Learn how to [generate a GPG key](#) and [add it to your account](#).

New GPG key

## Vigilant mode

☐ Flag unsigned commits as unverified



# GitHubに 公開鍵を設置

Github上のキー名は自由

今回はbusiness-pc-key

C:¥User¥ユーザー名¥.ssh

フォルダ内のid\_rsa.pubを

メモ帳などで開き、

Key 内に張り付けて保存する。

## 疎通確認

ローカルのgitBashで

ssh -T git@github.comと入力

パスワード等を入力し

接続okになることを確認する

A faint, light blue world map is visible in the background of the slide, centered behind the text.

# GitHub上で リモートリポジトリを作成

# GitHubでリモートリポジトリ

git-sampleなどでリポジトリを作成

```
git init
```

```
git add README.md
```

```
git commit -m "first commit"
```

```
git branch -M main
```

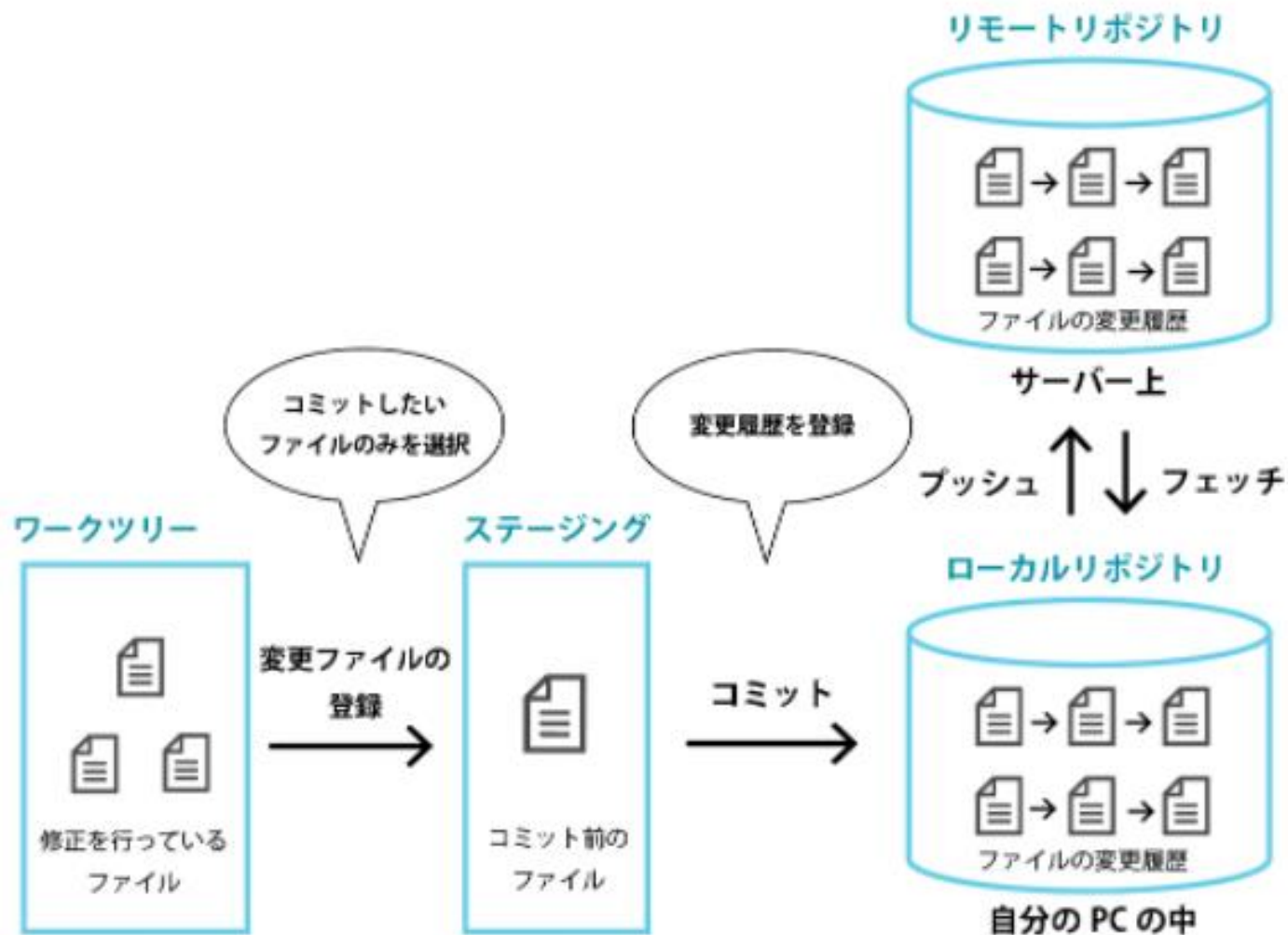
```
git remote add origin git@github.com:ユーザー名/git-sample.git
```

```
git push -u origin main
```

A faint, light blue world map is visible in the background of the slide, centered behind the main text.

# よく使うコマンド

# よく使うコマンド



# よく使うコマンド

git add . // 追加

git commit -m “コメント” //セーブ

git push origin main

// アップロード ローカル(origin) リモートブランチ

# ブランチの操作

git branch // ブランチ一覧表示

git branch <new> // ブランチ作成

git branch -m <old> <new> // ブランチ名変更

git checkout <name> // ブランチ切り替え

git checkout -b <new> // ブランチ作って切り替え

git switch -c <new> // ブランチ作って切り替え



# 状況確認

git status // コミット対象のファイル確認

git log // 全コミットログ

git config // 設定確認

git --version // バージョン確認

## 変更点の確認

git diff // 変更点を確認

git diff <commit\_id> <commit\_id>  
<name\_of\_file> // 2コミット間の差分

git fetch // リモートの情報をローカルにダウンロード

## ダウンロード関連

git clone remoteブランチ名 //丸ごとダウンロード

git pull origin main // 差分ダウンロード

# コミットやaddを取り消す

```
git reset <name_of_file>
```

```
git reset <commit_id> <name_of_file>
```

```
Git reset --soft <commit_id>
```

オプション

--soft コミットだけをなかったことにする

--hard ファイルの変更自体をなかったことにする(現在のファイルの中身も変わる)



# ブランチのマージ

```
git merge <name_of_branch>
```

## 変更点の一時保存

git stash list // stashのリストを確認

git stash save // 変更点をstashに保存

git stash pop // スタッシュを復元(スタッシュは破棄)

git stash apply // スタッシュを復元