

cs-15. 全体まとめ


(コンピューターサイエンス)

URL: <https://www.kkaneko.jp/cc/cs/index.html>

金子邦彦



謝辞：この資料では「いらすとや」のイラストを使用しています

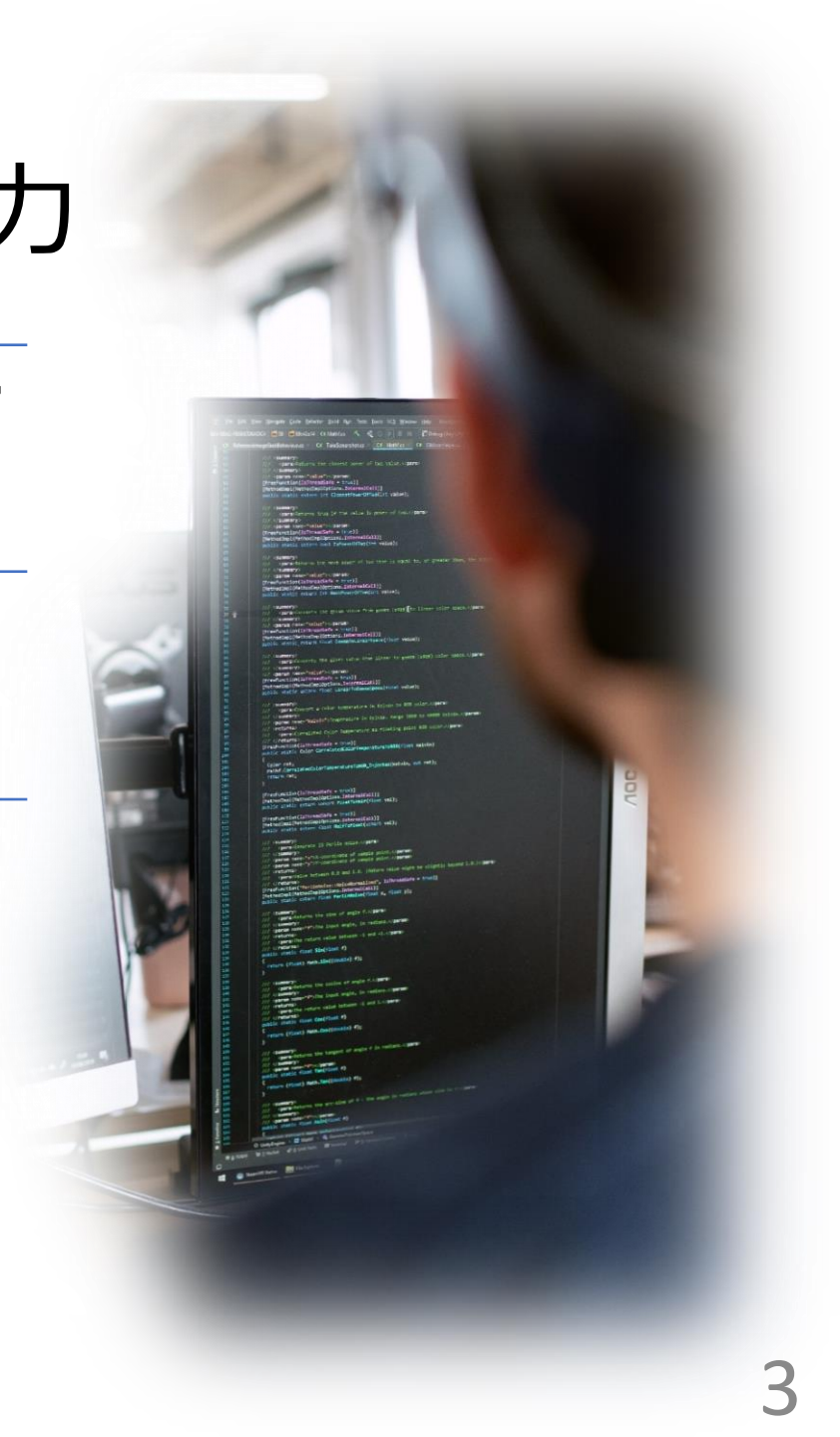
- 
- ① **情報工学の基礎から最新技術まで幅広くカバー．視覚的な説明**
 - ② **デジタルの基礎、プログラミング、データベース、人工知能など、情報工学の主要分野を振り返る．具体例をメインに説明．実践力アップ．**
 - ③ **プログラミングの基礎，AIツールの活用などの実践的な技術スキルに注目**
 - ④ **未来展望：AI，クラウドコンピューティング，IoTなど、将来，最先端技術を学ぶ時のための基礎と入門**

情報工学の3つの魅力

自分のアイデアを形にできる楽しさ

AI, 仮想現実, IoT など、未来の
技術のワクワク感

IT企業、製造業など幅広い分野で
活躍し、夢を追求できる



情報工学はデジタル社会の発展に欠かせない

- ・オンラインでの**交流・コミュニケーション**
- ・動画配信、オンラインショッピングなどの**デジタルサービス**
- ・**膨大な情報**の管理・処理
- ・**人間とAIの協働**
- ・自動運転、医療、教育など様々な分野での**AI活用**

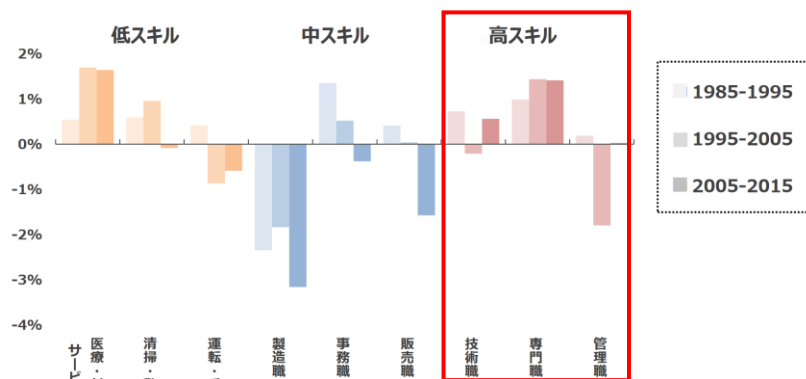


2050年の未来予測と情報工学

経済産業省のレポート「未来人材ビジョン」（令和4年5月）によると

<https://www.meti.go.jp/press/2022/05/20220531001/20220531001.html>

日本における職業別就業者シェアの変化



高スキル職は増加

「職種」… 事務従事者 42% 減少

販売従事者 26% 減少

情報処理・通信技術者 20% 増加

開発・製造技術者 11% 増加

2050年の未来ビジョン

- デジタル化の継続
- 脱炭素
- AIやロボットとの共生社会
- 日本の生産年齢人口の減少

情報工学科で専門的な知識とスキルを身につけることで、未来のITエンジニアという高スキル職で活躍し、社会に貢献しながら自分の夢を追求することができる

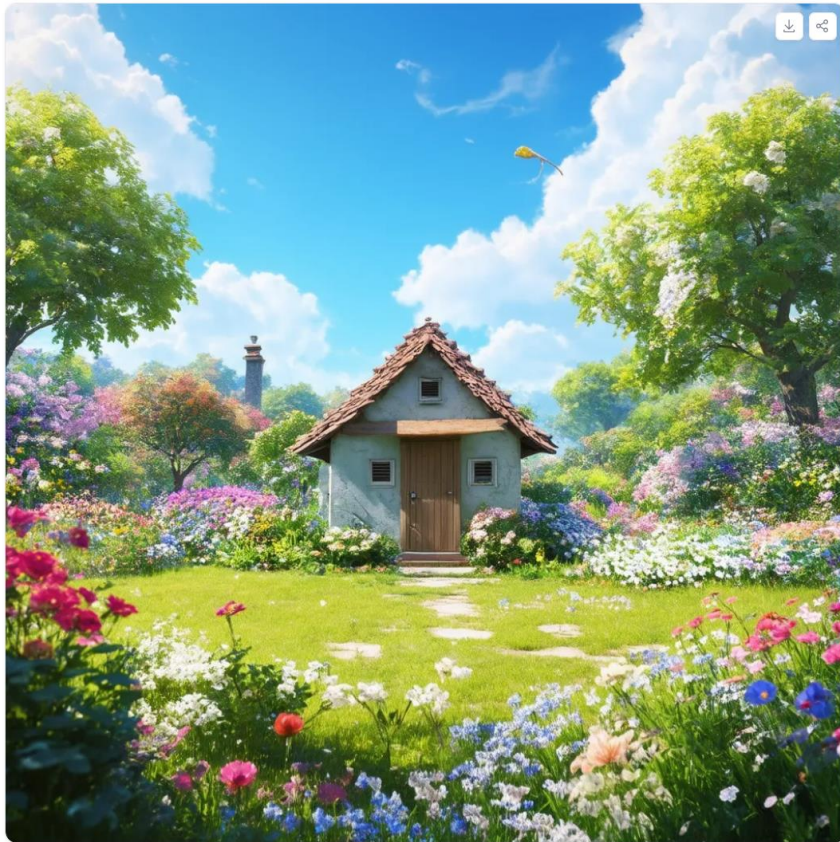
15-1 AIによる画像生成

演習 文章からの画像生成

ページ 6, 7

【トピックス】

- ・ 文章からの画像生成
- ・ プロンプト



Demo Stable Diffusion 3 Medium
を試す

<https://huggingface.co/spaces/stabilityai/stable-diffusion-3-medium>

プロンプトを**英語**で入れて
「**Run**」をクリック実行

思い通りの結果を得るためにプロ
ンプトを工夫する.

beautiful garden, small house,
many flowers, blue sky, clouds,
realistic, cinematic, landscape
vista photography, Ghibli

15-2 チャットボット（AIとの 対話，コミュニケーション）

演習 チャットボット

ページ 10, 11

【トピックス】

- ・ 生成AIに作業を頼む
- ・ 生成AIからアイデアや知識を引き出す

次は、チャットボットのChatGPT 3.5

<https://chatgpt.com/>

注意点：

- 秘密にしたい情報を投稿してはいけません.
- AIの回答は完璧に正確というわけではありません.

ChatGPT

- AI による対話システム
- 文章処理（要約、推敲、翻訳）のほか、問答により学び、新たな視点を得ることができる
- URL: <https://chat.openai.com/>
- 登録必要
- **AIの回答が間違っている場合がある**（必ず根拠を確かめよう）
- **秘密の情報やプライバシーについて投稿してはいけない**
- 大学のレポートなどは、学生自身の成果物を求めているので、AIの回答をそのまま提出してはいけない

ChatGPT との問答の例① 質問

プログラミングで if と for を習いました。他の重要なキーワードは？

プログラミングにおいて重要なキーワードはたくさんありますが、特に基本的なものをいくつか挙げると以下ようになります：

1. **while**: 条件が真の間、繰り返し処理を行うループ構造です。
2. **function** (関数): 名前付きのブロックで、特定の処理をまとめたものです。再利用性やコードの構造化に役立ちます。
3. **variable** (変数): 値を格納するための記号であり、プログラム内でデータを管理するために使用されます。
4. **array** (配列) や **list** (リスト): 複数のデータを1つの変数にまとめて管理するデータ構造です。
5. **class** (クラス): オブジェクト指向プログラミングにおいて、データとその操作を定義した設計図です。
6. **method** (メソッド): クラスに属する関数で、そのクラスのインスタンスに対して操作を行うために使用されます。
7. **module** (モジュール): 複数の関数やクラスをまとめたもので、再利用可能なコードのパッケージです。
8. **library** (ライブラリ): 汎用的な機能やデータ構造を提供するコードの集まりで、プログラミングの効率化や機能拡張に用います。
9. **exception** (例外): プログラムの実行中に発生するエラーや異常状態を処理するための仕組みです。

ChatGPT との問答の例② プログラム作成を頼む

次のプログラムを工夫して、花の形を書きたい

```
import turtle
turtle.Screen.clear()
t = turtle.Turtle()
t.shape("turtle")
t.goto(0, 100)
t.goto(100, 0)
```

次のプログラムを工夫して、花の形を描くことができます。以下は、Turtle Graphicsを使用して花の形を描く例です。

python

📄 コードをコピーする

```
import turtle

# 画面の準備
screen = turtle.Screen()
screen.bgcolor("white")

# Turtleの設定
t = turtle.Turtle()
t.shape("turtle")
t.speed(2) # 描画速度を設定

# 花を描く
for _ in range(36): # 36回繰り返すことで花を形成する
    t.circle(50) # 半径50の円を描く
    t.left(10) # 左に10度回転する

# 終了処理
turtle.done()
```

このプログラムでは、Turtleを使用して円を描くことで花の形を表現しています。`t.circle(50)`によって半径50の円を描画し、`t.left(10)`によって次の円を描く方向を左に10度ずつ回転させています。これにより、花の形を形成することができます。

実行すると、花の形を描いたウィンドウが表示されます。

ChatGPT との問答の例③ ミスを探すことを頼む

次のレポートは、どこかに誤字がありますか？ $100 * 1.1$ と $200 * 1.1$ という2つの式を抽象化すると $a * 1.2$ になる

はい、レポートに誤字があります。正しくは次のように修正する必要があります：

情報化社会

「 $100 * 1.1$ と $200 * 1.1$ という2つの式を抽象化すると $a * 1.1$ になる」

つまり、100と200の部分が誤って1.1となっているため、正しい数値は1.1ではなく1.0になる

ChatGPT との問答の例④ 疑問に思ったことを相談する

Python をパソコンで勉強するとき、最初の行動は何？

Pythonをパソコンで勉強するときの最初の行動は、Pythonのインタプリタ（対話型シェル）を起動することです。以下の手順で行います：

1. **Pythonのインストール:** もしパソコンにPythonがインストールされていない場合は、公式ウェブサイト（<https://www.python.org/downloads/>）から最新バージョンをダウンロードしてインストールします。
2. **コマンドプロンプトやターミナルを開く:** Windowsではコマンドプロンプト、MacやLinuxではターミナルを開きます。
3. **Pythonインタプリタを起動する:** コマンドプロンプトやターミナルで、`python` または `python3` と入力して、Pythonの対話型シェルを起動します。例えば、以下のようにします。

📄 コードをコピーする

```
python
```

もしくは

📄 コードをコピーする

```
python3
```

4. **Pythonインタプリタの動作確認:** Pythonインタプリタが起動すると、プロンプト (`>>>`) が表示されます。これはPythonがコマンドを受け付けている状態です。
5. **基本的な動作の確認:** 簡単な計算（例: `2 + 2`）を入力して実行し、Pythonが正常に動作していることを確認します。

Pythonのインタプリタを起動することで、Pythonの基本的な動作やコードの実行を確認しながら、学習を進めていく準備が整います。

ChatGPT との問答の例⑤ 勉強用に要点の作成を頼む

Python の基礎を 7 行で正確に簡潔にまとめてください

Pythonの大切な基礎をやさしく簡潔にまとめると次のようになります：

1. **シンプルで直感的な文法:** Pythonは読みやすく書きやすい言語です。
2. **変数とデータ型:** 変数は宣言せずに使い、動的に型が決まります（整数、浮動小数点数、文字列など）。
3. **制御構造:** if文（条件分岐）、forループ（繰り返し）、whileループ（条件繰り返し）を使ってプログラムの流れを制御します。
4. **関数とモジュール:** 処理をまとめた関数を定義し、再利用可能にします。さらに、標準ライブラリやサードパーティのモジュールをインポートして機能を拡張します。
5. **データ構造:** リスト（順序付きコレクション）、タプル（不変なリスト）、辞書（キーと値のペア）など、さまざまなデータ構造を使ってデータを効率的に管理します。
6. **エラー処理:** try-except文を使ってエラーをハンドリングし、プログラムの安定性を確保します。
7. **オブジェクト指向:** クラスとオブジェクトを使ってデータと処理を一体化し、コードの再利用性と保守性を高めます。

これらの基礎を理解することで、Pythonを使ったプログラミングの基本的な概念と技術を身につけることができます。

ChatGPT 利用上の注意点を再確認

- **AIの回答が間違っている場合がある:**

AI は、不正確な回答を提供する可能性がある。提供される回答は、参考にとどめ、**必ず他の信頼性の高い情報源などを確認**する。

- 秘密の情報やプライバシーについての投稿:

個人情報や機密情報を投稿しないこと

- 自習などでの利用:

AIは、相談相手として、自習などに役立つ。しかし、**AI の回答をそのまま学校のレポートや宿題として提出してはいけません**。レポートや宿題は、学生自身の知識と理解を高めるためのもの。AIは参考として活用し、自分自身の考えや研究成果など、自分の成果物を提出することが大切。

15-3 デジタルの基礎

ビットとデジタル化



- コンピューターは、すべての情報を0と1の組み合わせで表現（デジタル化）
- 一つの「0」または「1」は一ビット（Bit）と呼ばれる
- ビットは、情報の最小単位

0011010111101110101011 . . . 23ビットのデータ

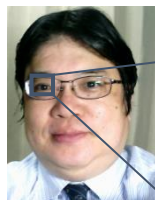
文字はデジタルの2進数に変換

- **ASCII**は**文字情報を数値で表現するためのもの**
- 1つの文字を, **7ビット**で表現
- 英数字や記号、制御文字など**128種類の文字**を表現することができる

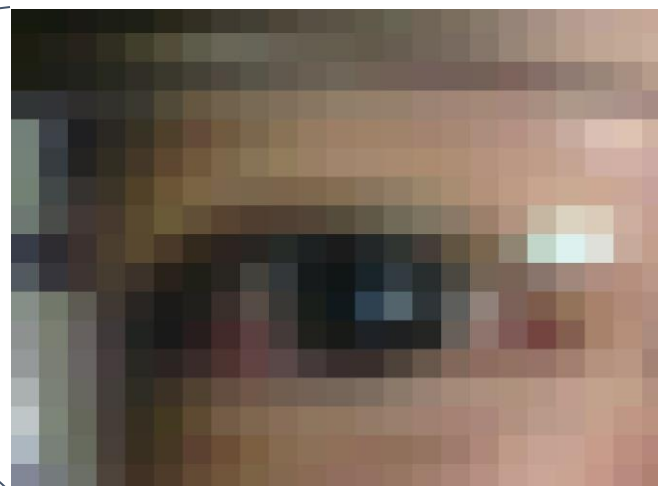
	0	1	2	3	4	5	6	7
0	NULL	DEL	SP	0	@	P		p
1	SOH	DC1	!	1	A	Q	a	q
2	STX	DC2	"	2	B	R	b	r
3	ETX	DC3	#	3	C	S	c	s
4	EOT	DC4	\$	4	D	T	d	t
5	ENQ	NAK	%	5	E	U	e	u
6	ACK	SYN	&	6	F	V	f	v
7	BEL	ETB	'	7	G	W	g	w
8	(BS)	CAN	(8	H	X	h	x
9	(HT)	EM)	9	I	Y	i	y
A	(LF)	SUB	*	:	J	Z	j	z
B	(VT)	ESC	+	;	K	[k	{
C	(FF)	(FS)	,	<	L	¥	l	
D	(CR)	(GS)	—	=	M]	m	}
E	SO	(RS)	.	>	N	^	n	~
F	SI	(US)	/	?	O	—	o	DEL

文字の7ビットを
16進数2桁で
示した表

画像と画素



画像



それぞれの格子が画素

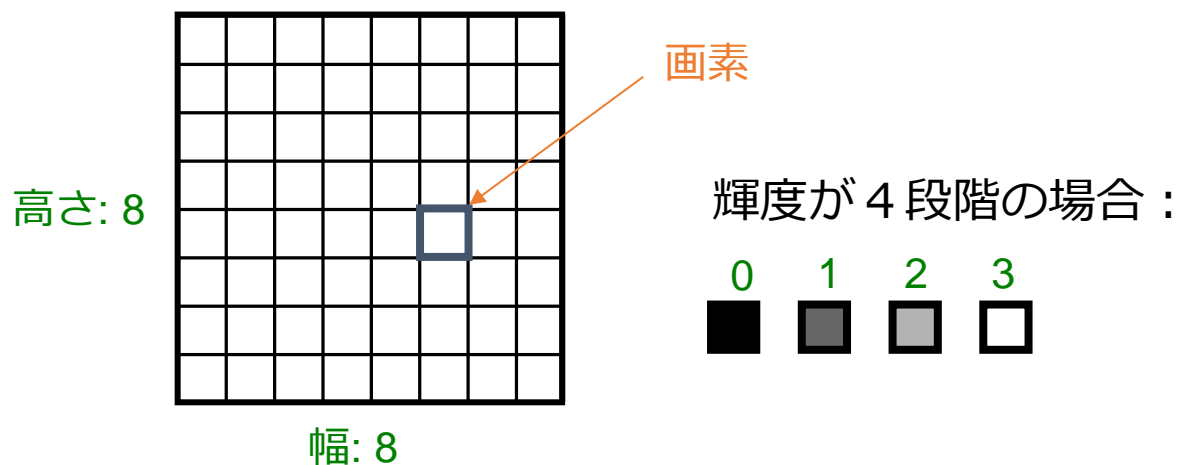
濃淡画像でのコード化



画像の輝度の情報

例えば： 黒 = 0 ,
暗い灰色 = 1 ,
明るい灰色 = 2 ,
白 = 3

のように**コード化**



カラー画像の成分

**R（赤）成分， G（緑）成分， B（青）成分で考える
場合**



R（赤）成分



G（緑）成分



B（青）成分

デジタル画像

各成分での「画素の明るさ」が数値化される

- コンピューターは、すべての情報を**0と1の組み合わせ**で表現（**デジタル化**）
- 文字、数値、画像、音声など、すべてが2進数に変換されている。

15-4 プログラミングの基礎

コンピュータとプログラム



- **コンピュータ**は、**プログラム**に従って動作
- **プログラム**は、**コンピュータ**に指示を出し、所定の作業を遂行させる

```
x1="100%" y1="0%" x2="0%" y2="100%"
color="#06101F" offset="25"/>
color="#1D304B" offset="100"/>
```

```
height="450" rx="8" fill="white" style="border: 1px solid black; border-radius: 8px; padding: 10px; margin-bottom: 10px;">
```

```
height="96" viewBox="0 0 96 96" style="display: inline-block; vertical-align: middle;">
```

```
gradient x1="87.565%" y1="15.173%"
```

```
Gradient>
```

```

op stop-color;
rGradient>
x="-500%" y="-500%" width="1000%"
dy="16" in="SourceAlpha"

```

プログラミング



- **プログラム**を設計し作成する
プロセス（プログラミング）
は、**創造的な活動**
- アイデアを形にできることが、
プログラミングの魅力



プログラム



- プログラムとは、コンピュータに実行させたい一連の手順を記述したもの
- プログラムの**起動**により、手順を実行し、必要な処理を行う



```
In [7]: from keras.models import Sequential
...: model = Sequential()
...: from keras.layers import Dense, Activation
...:
...: model.add(Dense(units=64, input_dim=len(x_train[0])))
...: model.add(Activation('relu'))
...: model.add(Dense(units=max(set(y_train)) - min(set(y_train)) + 1))
...: model.add(Activation('softmax'))
...: model.compile(loss='sparse_categorical_crossentropy',
...:               optimizer='sgd',
...:               metrics=['accuracy'])
...: model.fit(x_train, y_train, epochs=200)
...: score=model.evaluate(x_test, y_test, batch_size=1)
...: print(score)
...: model.predict(x_test)
...: model.summary()

Epoch 1/200
3/3 [=====] - 0s 5ms/step - loss: 1.0583 - accuracy:
0.3200
Epoch 2/200
3/3 [=====] - 0s 0s/step - loss: 1.0530 - accuracy:
0.3200
Epoch 3/200
3/3 [=====] - 0s 0s/step - loss: 1.0485 - accuracy:
0.3200
```

Python 言語を使って
ニューラルネットワーク
を作成. AIシステムを構築

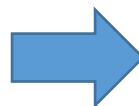
プログラムは, **コンピュータ**の動作を細かくコントロール

プログラミングの目的



- **プログラム**は, **コンピュータ**に指示を出し, 所定の作業を遂行させる
- 複雑な作業も**自動化**し, 効率化することが可能

```
a = [200, 400, 300]
for i in a:
    print (i * 1.08)
```



```
216.0
432.0
324.0
```

Python プログラムの
ソースコード

プログラムの
実行結果

プログラミングで、自分のやりたいことを実現するために



• AI支援

ChatGPTなどのAIチャットボットを使って、疑問点を質問したり、コードの説明を求めたりできる。

• 良いツールを自分で探求する

- JupyterLab : Python によるデータ分析, AIの試行ができる. プログラムと結果を一つのページでまとめて管理できる
- Visual Studio Code : 豊富な拡張機能、ビジュアルな操作
- Google Colaboratory : インストール不要, Python の AI 活用が容易な環境

• YouTube 動画, オンライン学習サイト (trinket など) の活用

• ビジュアルプログラミングの Scratch

(<https://scratch.mit.edu/>) ブロックを組み合わせてプログラミングを学ぶ

- **プログラム**は, コンピュータに指示を与えるもの
- 複雑な作業も自動化できる

15-5 データベースの普及

データベース

データベースは、特定の主題について**整理**、**保存**、**管理**された**データ**の集合体



取引



記入

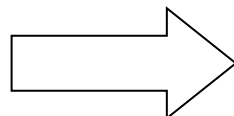


計測

撮影



データ保存



データ収集



データベース

データベースの利用分野

①オンラインコミュニケーション

オンラインコミュニケーションでの
データ管理。機能がより便利に。

- **SNS（ソーシャルネットワーク）**

投稿、ユーザプロフィール、「いいね」、コメントの管理

- **電子メール**

本文、添付ファイル、送信者、受信者の管理

- **オンラインのチャット**

ユーザ間のメッセージの管理



データベースの利用分野

②オンラインの取引

リアルタイムで安全、便利なサービスの提供.

- ・ **オンラインの取引**

注文, 支払い, 配送状況問い合わせ

- ・ **オンラインの銀行**

送金, 残高照会, 融資申請

- ・ **オンラインの予約**

列車や飛行機などの座席予約

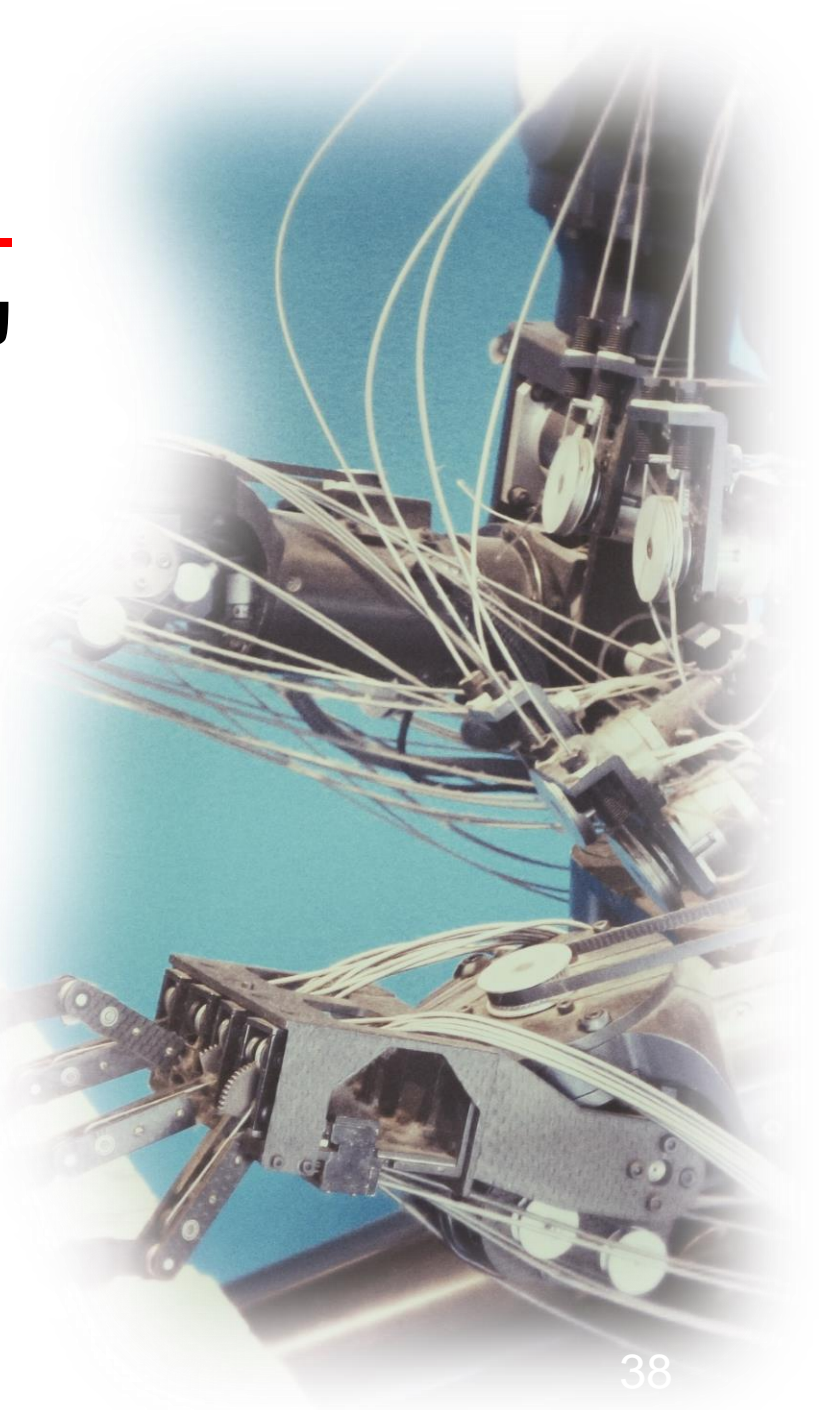


データベースの利用分野

③人工知能

人工知能での学習による上達：**データを使用**し，**学習**を通じて**知的能力**を向上.

- ChatGPT などの対話型AI
(対話，自由なアイデア出し，
要約，翻訳など)
- 医用画像や自動運転での画像理解
(画像診断、物体認識など)
- オンラインショッピングでの情報
推薦
(過去の履歴からの商品の順位付
けなど)



データベースシステム

- 膨大なデータを整理し，必要な時にアクセスできるようにする「**データベースシステム**」
- オンラインコミュニケーション，オンライン取引，人口知能（AI）など，社会を支える基盤である

15-6 人工知能

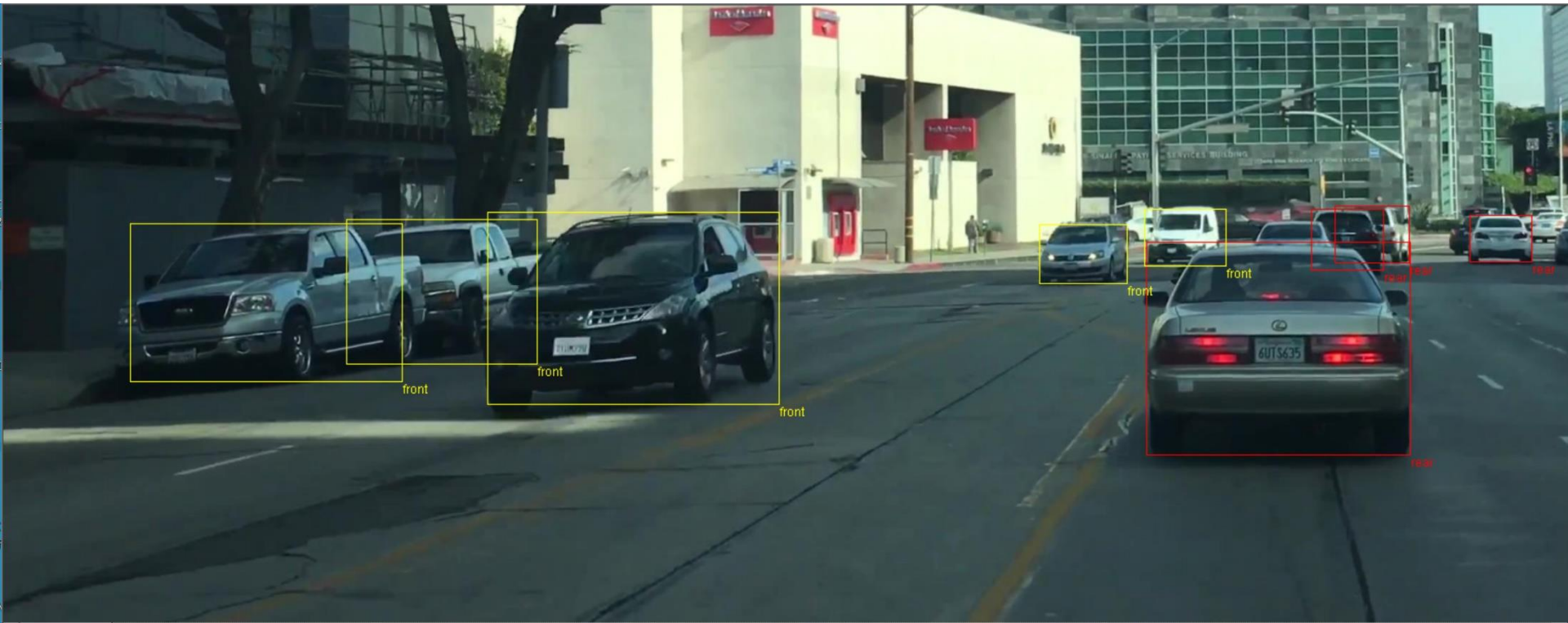
- ・人工知能は、コンピュータが知的な能力を持つこと

知能：思考や判断などの能力

知識：情報を扱う能力

学習：知的な能力が上達できる能力

車両の発見・検知



人工知能は、車両の場所と**向き**（前なのか後ろなのか）を素早く発見できるようになってきた
（Dlib を使用）

人や自転車などの，オブジェクトの発見・検知



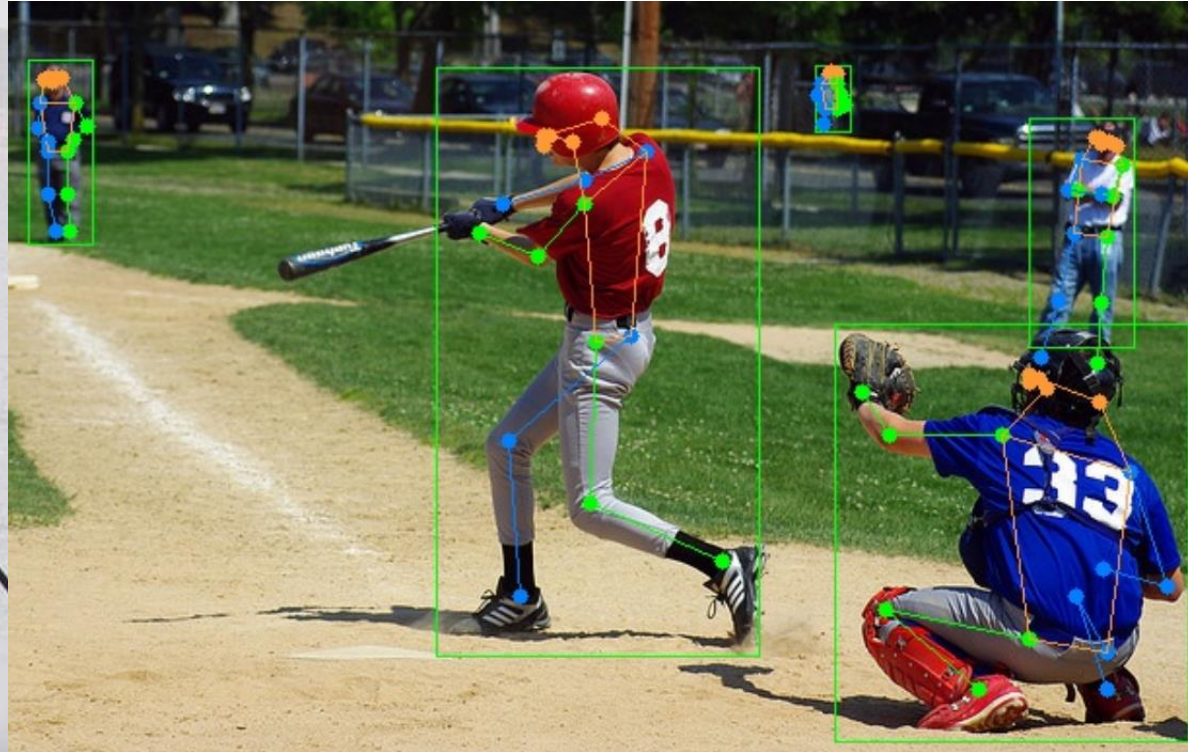
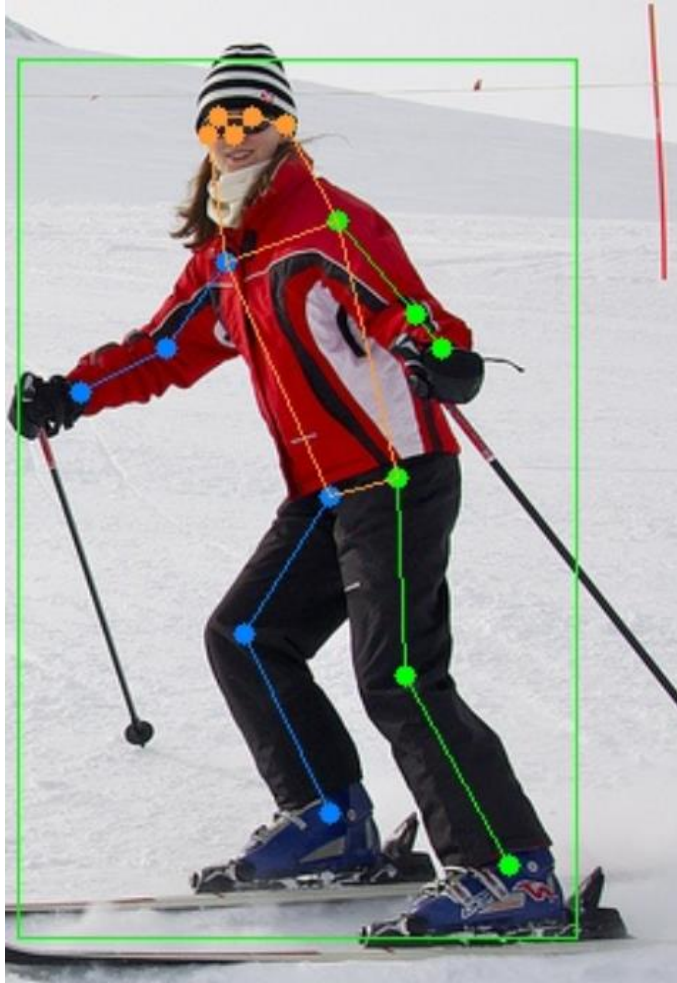
元画像



人工知能による読み取り結果
(DeepLabv3+ を使用)

- 人間の「目」の一部機能をコンピュータで再現.
画像の中のオブジェクトを，**人工知能**が発見・検知

人体の向き，ポーズの読み取り

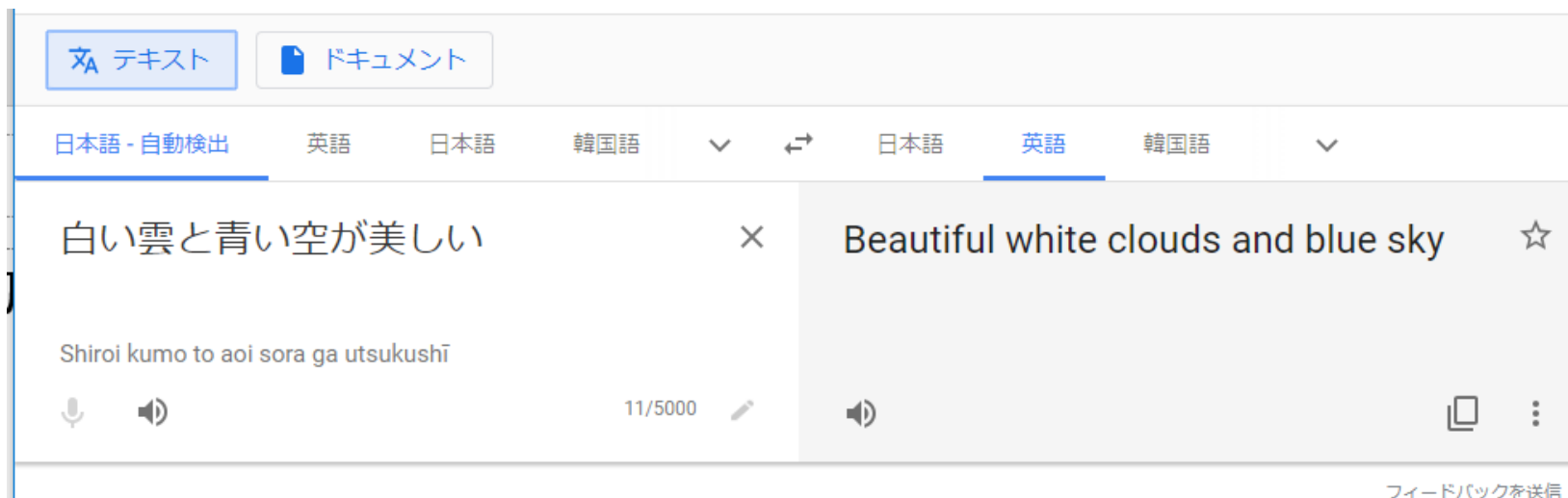


- 写真やビデオから，人体の姿勢を読み取り

翻訳を行うオンラインサービス



Web ブラウザで動く



DeepL の URL: <https://www.deepl.com/ja/translator>

- 人工知能（AI）の究極の目標は、コンピュータで人間の知能を模倣すること
- 学習、問題解決、パターン認識、対話などを行う
- AIは、すでに、私たちの日常生活に深く浸透している

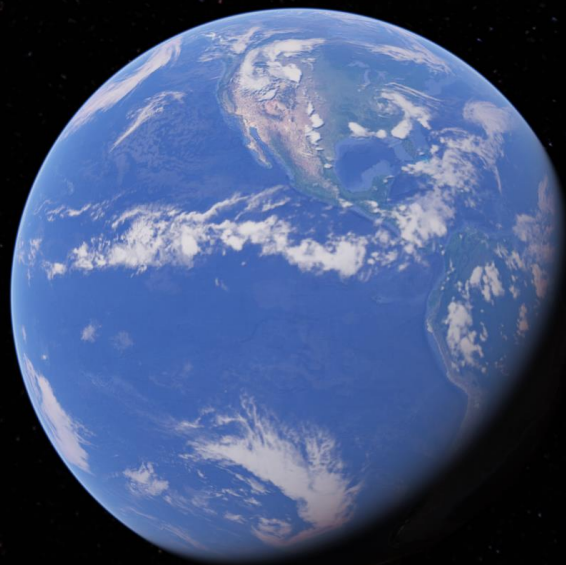
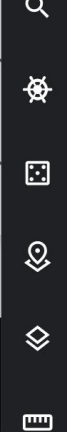
例： スマートフォンの音声アシスタント
自動運転車

- 生活と社会が大きく変化

例：

- 医療診断，個別化医療の実現
- 自動運転技術，交通事故の減少
- 言語の壁を越えたコミュニケーションの促進

15-7 3次元コンピュータグラフィックス



Google Earth の起動

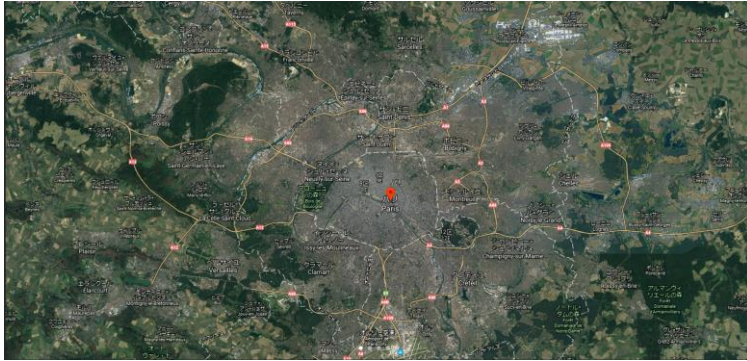
Google Earth は、**無料で利用可能なオンラインの地球儀**

① ウェブブラウザで次の URL を開く

<https://earth.google.com>

② 「**Earthを起動**」をクリック

Google Earth の機能



写真



3次元コンピュータグラフィックス

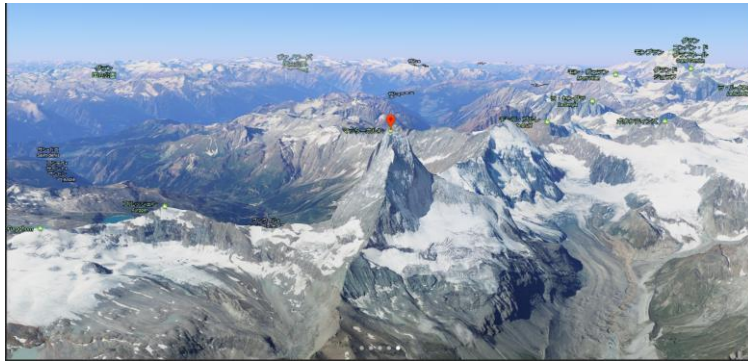


写真 + 標高



地球儀



ストリートビュー



Google Earth で「福山城」を検索したところ

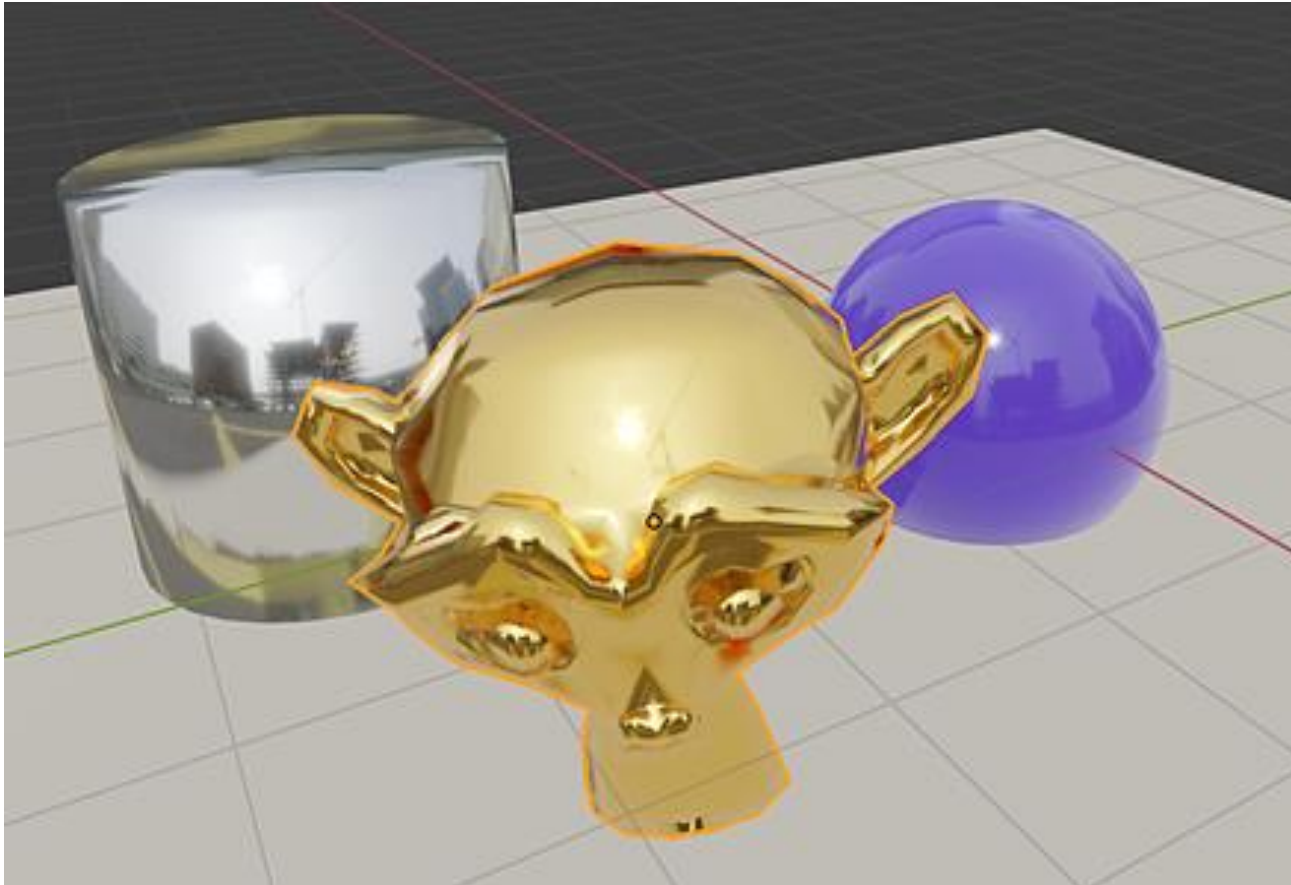


視点移動は，マウス操作でできる

3次元コンピュータグラフィックスソフトウェア

Blender でできること

3次元のリアルな造形



Blender のPBR マテリアル

PBRマテリアルを使うと，このようなことも可能

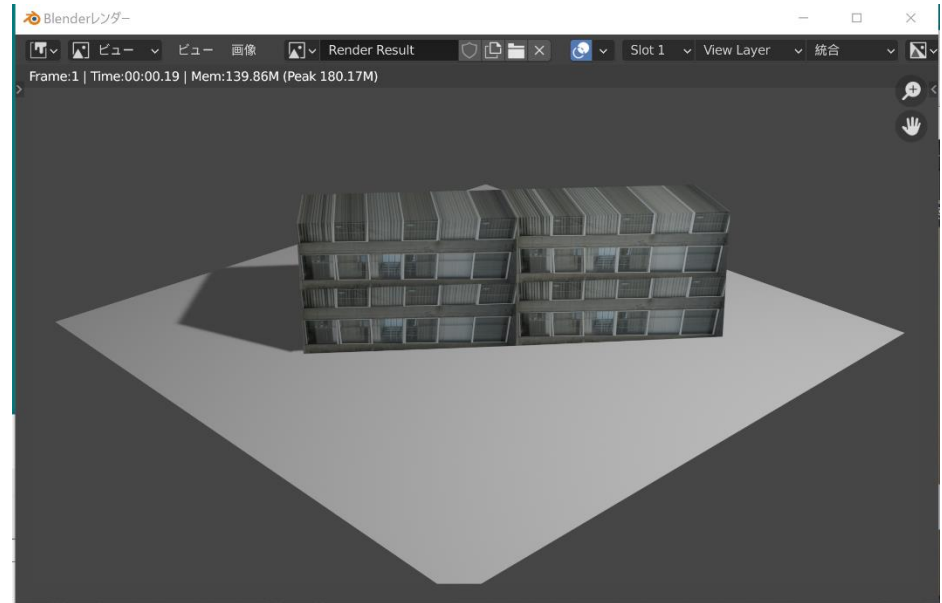
3次元コンピュータグラフィックスソフトウェア

Blender でできること

3次元のリアルな造形



元画像

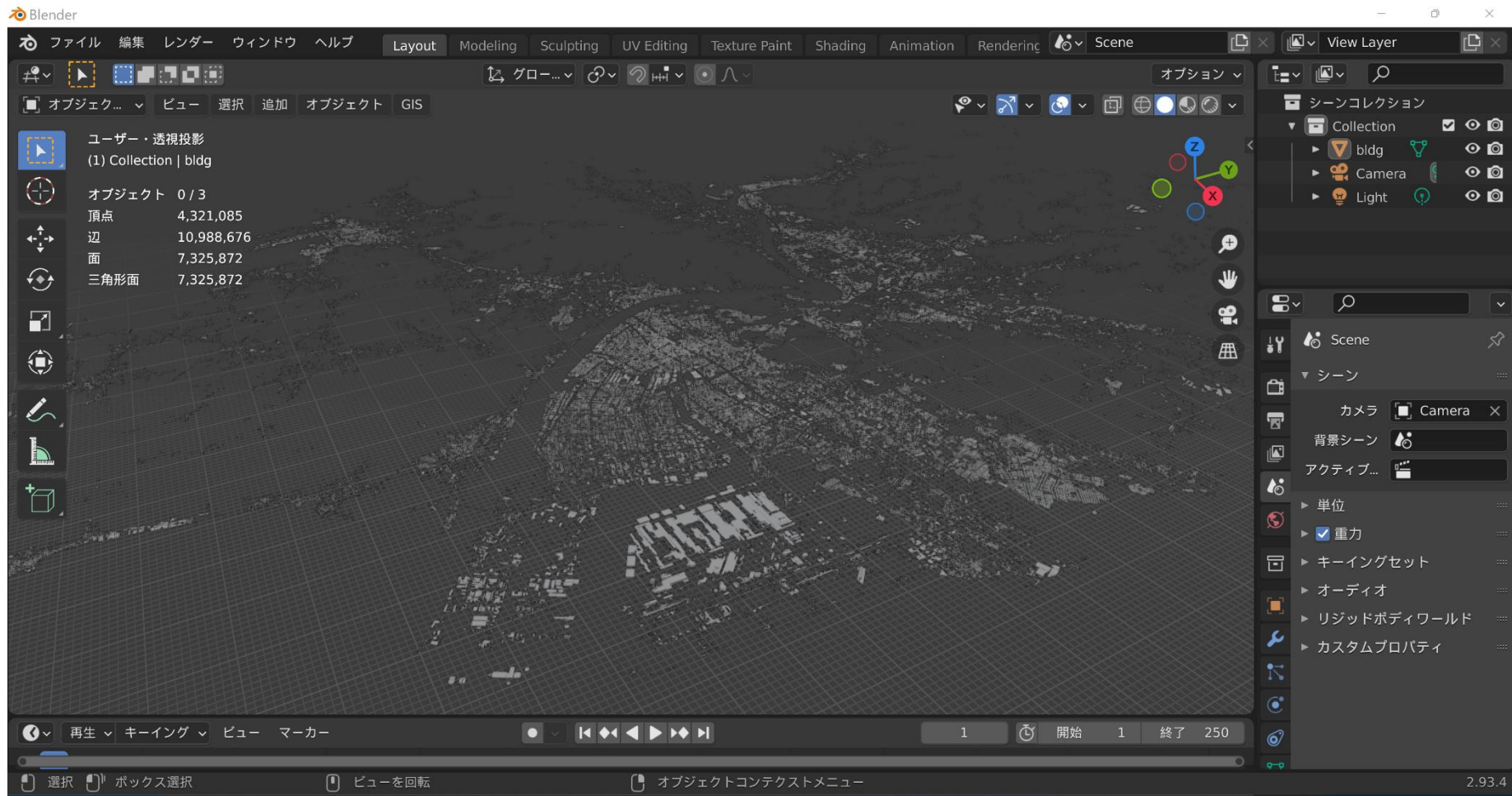


Blender での
テクスチャマッピング

3次元コンピュータグラフィックスソフトウェア Blender でできること



仮想空間の実現



3D都市モデル（Project PLATEAU）福山市（2020年度）
建物のデータ頂点数: 4,321,085, 三角形ポリゴン数: 7,325,872

3次元コンピュータグラフィックスと仮想世界



- 3次元コンピュータグラフィックスにより，コンピュータ上に，現実世界の再現，新しい世界の造形が可能になった
- ゲーム，建築，医療など、様々な分野で活用
- Blender などのソフトウェアの普及．一般のパソコンでも可能．

15-8 情報セキュリティの重要性

情報の広範な流通と共有

情報の価値向上 **デジタル社会** 個人が情報発信者に

技術革新による社会変革

フェイクビデオ



金子の顔

+



有名人の声，表情，
語り

→



金子がその有名人
そっくりで語りだす

- AI の進化により，映像や音声の偽造が容易となり，それが事実かどうかの判断が難しくなっている

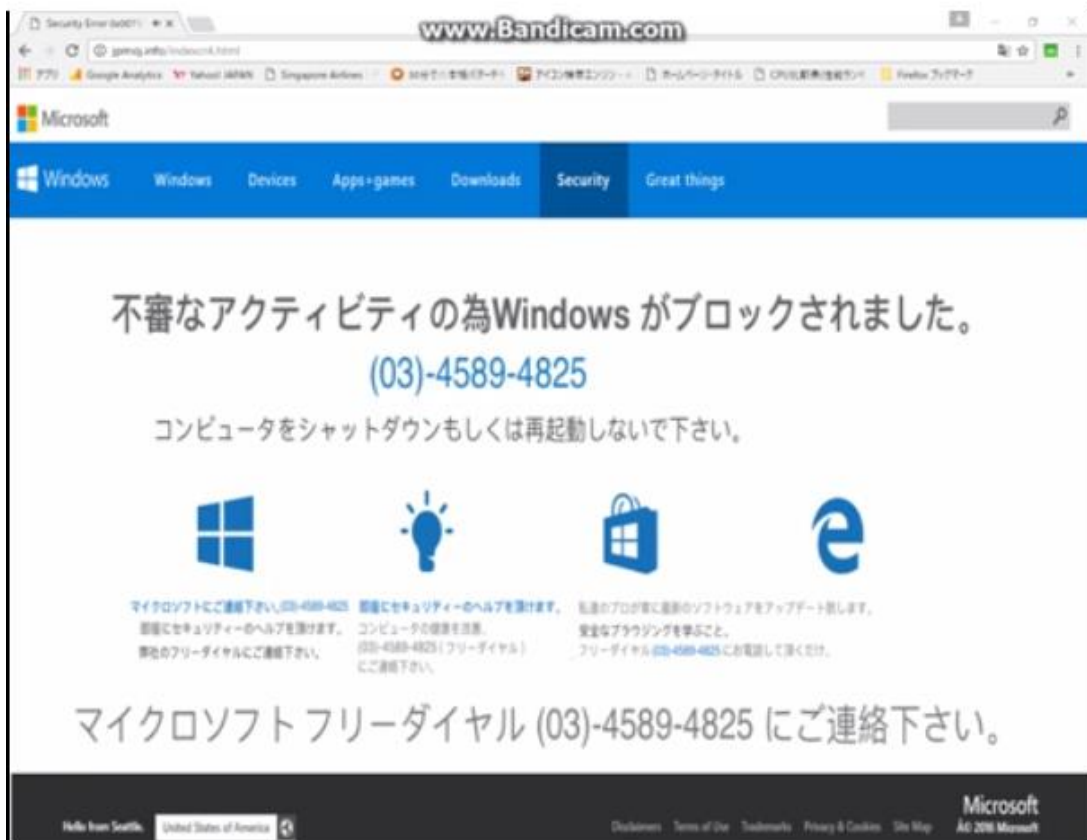


電話連絡！

送金！

**ウェブサイトへの
アクセス！**

**を求める詐欺行為
がある**



**個人情報保護とフィッシング詐欺から身を守るために、
怪しいメールやウェブサイトへのリンクをクリックしない
ように注意が必要です**

有害なソフトウェア（マルウェア）の仕組み



- 侵入：メールの添付ファイルや不正サイトからダウンロード
- 感染：システムに潜伏
- 活動：個人情報の窃取、システムの破壊など
- 拡散：他のデバイスやネットワークに感染を拡大

- **デジタル社会**において**情報セキュリティ**は必要不可欠である。
- **パーソナルな情報の保護**: 個々人の情報は重要な資産であり、各自が尊重し、守る責任がある。
- **不正アクセスの防止**: オペレーティングシステムのアップデート（更新）などの対策が大切である。
- **データバックアップ**: データの消失を防ぐために、定期的なバックアップを行うべきである。
- **社会的なルールの理解と遵守**: マナーを理解し、遵守することが重要である。

15-9 展望

- コンピューターサイエンスは、私たちの社会と生活を変える力を持つ
- 情報工学分野を学ぶことで「未来を創る」ための力を持つことができる



情報工学の未来技術（大学の授業で学ぶ）

人工知能（AI）, クラウドコンピューティング,
IoT（Internet of Things, もののインターネット）サイ
バーセキュリティ, 通信技術, 仮想現実 など

今回の授業の意義と満足感



- ① **総合的な技術知識**： AI、3次元グラフィックス、データベース、セキュリティなど
- ② **実践的スキルと問題解決力の向上**： **プログラミング、チャットボット活用などの実践的スキル**
- ③ **技術の社会的影響と倫理的視点**： 情報技術が社会や産業に与える影響、技術者としての責任を理解
- ④ **キャリア展望と創造的思考**： 技術革新がもたらす将来の可能性、未来社会の課題解決に向けた創造的思考力の向上