

2020年度春学期における 大阪大学の初年次一般情報教育

白井 詩沙香

大阪大学 サイバーメディアセンター

情報社会基礎・情報科学基礎

- **全学共通教育科目の一般情報教育科目**

- 全学教育推進機構を開講部局とし、サイバーメディアセンターがコースデザイン・教材・演習ツール等の提供を担当

- **春学期開講の1年生向け※必修科目**

- 週2コマの2単位科目

- **2019年度にカリキュラム改革を実施**

- 2018年度までは前期に各学部学科で独自の内容を実施
- 2019年度よりSociety5.0時代に向けた一般情報教育として学部共通の学習項目を設定し（IPJSJのGEBOKを参考）、授業内容をリニューアル
- 各学部学科の授業担当の先生方のご協力のもと授業を実施

本授業の特徴

反転学習的アプローチ

- 週2回の授業が**対面授業（同期）**と**オンライン授業（非同期）**から構成
※2020年度は全てオンライン授業（13回）
- LMSを通じて各種教材を提供。毎授業後はリフレクションを兼ねた授業アンケート
- 前半はオンライン授業回は講義動画による**知識習得型授業**・対面授業回は**演習授業**
- 後半の**プログラミング授業回※は全て演習**
(情報社会基礎：3回、情報科学基礎：5回)

➡ **どのような学習環境で演習を行うか？**

情報科学基礎の標準授業計画

回	トピックス
1	ガイダンス
2, 3	メディアとコミュニケーション
4, 5	情報のデジタル化とコンピューティングの要素と構成
6, 7	情報ネットワークと情報セキュリティ
8	中間テスト + 前半授業の振り返り
9~11	プログラミング演習
14	インターネットサービスの仕組み
15	期末テスト

2020年度の対応方針

プログラミング経験・学習環境（PCスペック、ネットワーク環境）も多岐にわたるため、**同期型授業回も非同期型授業回と同じく、個々のペースで学習を進めることができるよう教材・学習環境を用意**し、必要な時にすぐに支援できる体制（遠隔会議システム、LMSの掲示板等）を整える

前半の演習

LMSのクイズ・解説動画等を準備し、自律的に演習を進めていけるよう準備

後半の演習

SaaSの**プログラミング演習環境 Ed**を活用した演習を実施。

- ➡ 2019年度から非同期型に対応できるよう教材等は準備できていたため、2019年度の状況を踏まえ難易度の調整するとともに解説動画の準備

前半：演習授業



- 情報社会基礎 火1:白井 詩沙香 - 2020年度 - 全学教育推進機構
- ホームページ
- 授業コンテンツ
- 掲示板
- 成績表
- メール
- ヘルプ
- 所属グループ
 - 教員・TA
- コース管理
 - コントロールパネル
 - コンテンツ管理
 - コースツール
 - 利用状況
 - 成績管理

授業コンテンツ



情報社会基礎 授業コンテンツ

このページでは情報社会基礎の履修に必要なオンライン教材・オンラインクイズ等を提供しています。決められた日時に教材が表示されるので、受講生のみなさんは期限内に受講するようにしてください。受講スケジュールについては「はじめに（授業を受ける前に必ず読んでください）」を確認してください。



はじめに（授業を受ける前に必ず読んでください）

本授業の目的・授業形態・授業スケジュールについて説明します。受講前に必ず読むようにしてください。



画面共有サポートルーム

第9回目授業からHTML、第11回目授業からプログラミング演習が始まります。画面共有による個別対応が必要な際に、Blackboard Collaborate Ultraを利用します。

- マイクやカメラ接続に関する設定・トラブル対応は[こちらのマニュアル](#)を参照してください。
- 第9回目のBCUの録画を視聴するには、[BCU記録動画視聴方法.pdf](#)

を参照してください。

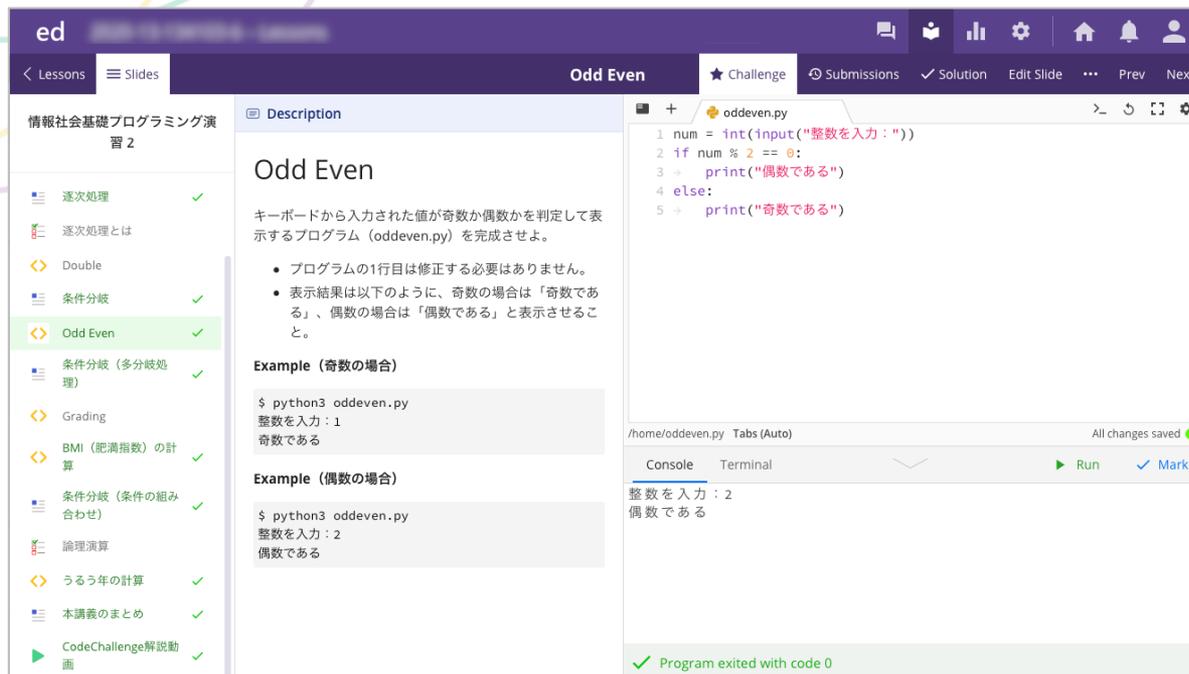


3. 情報とコミュニケーション（メディア授業）（4/21）

後半：プログラミング学習環境

Ed platform (Edstem.org)

- The University of Sydneyでも採用



The screenshot shows the Ed platform interface for a Python challenge titled "Odd Even". The interface is divided into several sections:

- Navigation:** A top bar with "Lessons" and "Slides" tabs, and a "Challenge" tab selected. Below it, a breadcrumb trail shows "Lessons" > "Odd Even" > "Challenge".
- Left Sidebar:** A list of lessons under the heading "情報社会基礎プログラミング演習 2". The "Odd Even" lesson is highlighted in green.
- Description Panel:**
 - Title:** "Odd Even"
 - Text:** "キーボードから入力された値が奇数か偶数かを判定して表示するプログラム (oddeven.py) を完成させよ。"
 - Instructions:**
 - プログラムの1行目は修正する必要はありません。
 - 表示結果は以下のように、奇数の場合は「奇数である」、偶数の場合は「偶数である」と表示させること。
 - Example (奇数の場合):**

```
$ python3 oddeven.py
整数を入力: 1
奇数である
```
 - Example (偶数の場合):**

```
$ python3 oddeven.py
整数を入力: 2
偶数である
```
- Code Editor:** A Python script named "oddeven.py" is shown with the following code:

```
1 num = int(input("整数を入力: "))
2 if num % 2 == 0:
3     print("偶数である")
4 else:
5     print("奇数である")
```
- Terminal:** Shows the execution of the script with input "2", resulting in "偶数である".
- Status:** A green checkmark at the bottom indicates "Program exited with code 0".

本授業では主に以下の3つの機能を活用

- **Ed Lessons**
授業教材・課題
- **Ed Discussion**
コミュニケーション
- **Analytics**
学習分析

後半：プログラミング学習環境

ed 2020-13-134103-6 - Lessons



Lessons

Search lessons

Grid

List

New Module

New Lesson



Lesson9: HTML演習



情報社会基礎プログラミング演習 1



情報社会基礎プログラミング演習 2



情報社会基礎プログラミング演習 3



情報科学基礎 プログラミング演習 4



情報科学基礎 プログラミング演習 5



Ed Lessons



ed

Lessons Slides 繰り返し処理：while文 Edit Slide Prev Next

情報社会基礎プログラミング演習 3

- 前回の内容 ✓
- 本講義の内容 ✓
- 繰り返し処理：while文 ✓
- 繰り返し処理とは
- 1から100までの和
- 10から200までの偶数の和
- 繰り返し処理：for文
- 整数sからeまでの和
- Blast off!
- 制御構造の組み合わせ
- FizzBuzz!
- アルゴリズムの基本
- 整列アルゴリズム
- 選択ソートとは
- 整列アルゴリズムの比較
- 整列アルゴリズムの考察
- 本講義のまとめ

繰り返し処理：while文

繰り返しは「条件を満たす間、処理を繰り返す」制御構造であり、Pythonでは「while」構文を使って記述します。また、変数を指定した値で変化させながら繰り返し処理を行う「for」構文があります。まずは「while」構文からみていきましょう。

while構文は、条件が真（True）の間、指定された処理を繰り返すものです。if構文と同様に、条件式に続く行に、インデント（字下げ）して処理を記述します。条件式には、if構文と同じく、比較演算子と論理演算子を用いた式を書きます。

では、1から3までの数字を画面に表示するプログラムを通して動作確認をしましょう。

Example：1から3までの数字を画面に表示する。

```

> Run PYTHON
1 i = 1
2 while i <= 3:
3     print(i)
4     i = i + 1
  
```

1行目で変数*i*にスタートの数字として1を代入しています。2行目から4行目がwhile構文で、2行目に変数*i*が3以下

1つの授業回の教材を Lessonsで開発

Reading document

テキストベースの教科書。
Code Snippetと呼ばれるリアルタイムに実行できるエディタを埋め込めるので、実際にプログラムを動かしながら学習を進めることができる。

Multiple Choice

Video

Code Challenge



Code Challenge

ed

Lessons Slides 1から100までの和 Challenge Submissions Solution Edit Slide Prev Next

情報社会基礎プログラミング演習 3

- 前回の内容 ✓
- 本講義の内容 ✓
- 繰り返し処理: while文 ✓
- 繰り返し処理とは
- 1から100までの和**
- 10から200までの偶数の和
- 繰り返し処理: for文
- 整数sからeまでの和
- Blast off!
- 制御構造の組み合わせ
- FizzBuzz!
- アルゴリズムの基本
- 整列アルゴリズム
- 選択ソートとは
- 整列アルゴリズムの比較
- 整列アルゴリズムの考察
- 本講義のまとめ

Description

1から100までの和

1から100までの整数の和を求めるプログラム (sum.py) を完成させなさい。

Example 1

```
$ python3 sum.py
5050
```

① 問題文

```
1 i = 1
2 sum = 0
3 while i <= 100:
4     sum = sum + i
5     i = i + 1
6 print(sum)
```

Console Terminal Run Mark

5050

**② 解答エリア
<プログラムの入力>**

実行ボタン **解答提出ボタン**

③ 実行結果

✓ Program exited with code 0

- プログラミング課題として利用
- 試行錯誤しながら課題に挑戦。自分の理解度をこまめにチェック。
- **自動採点**が可能
- 教員は受講生の画面をリアルタイムで確認・共同編集も可能

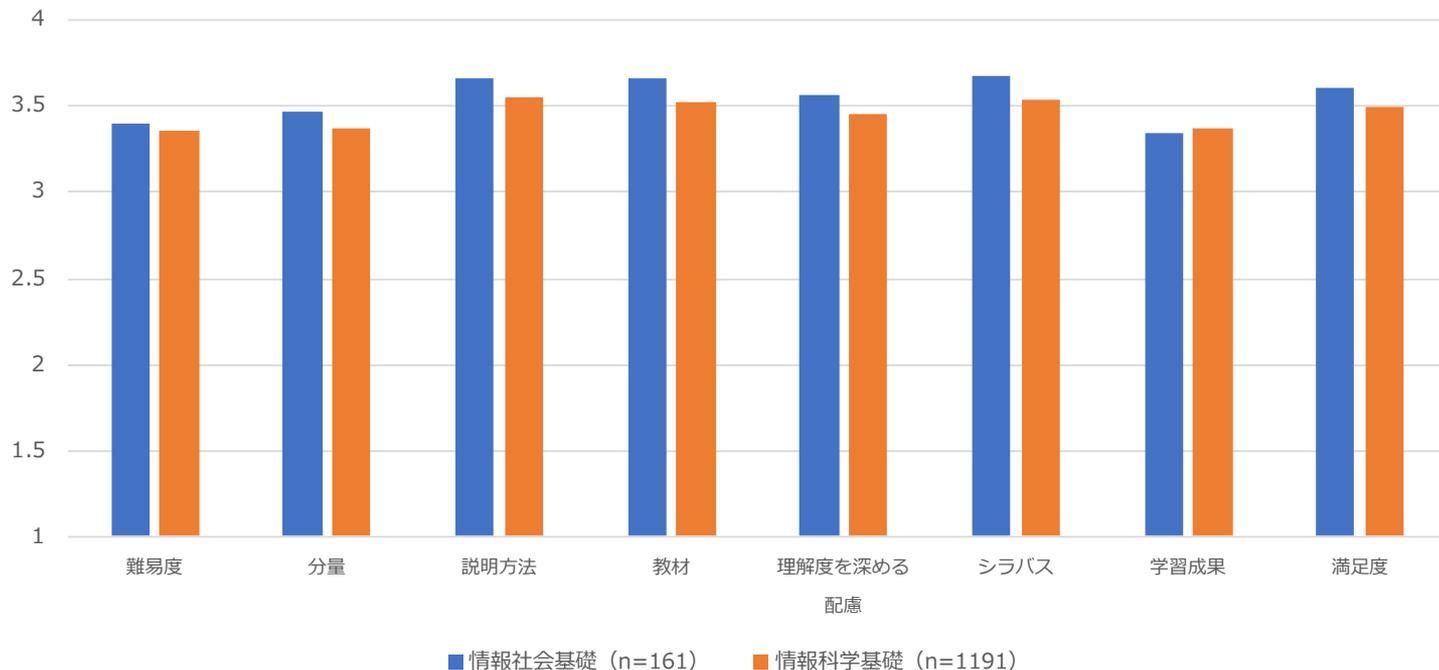
Analytics



- クラス・受講生単位で進捗状況の可視化
- Ed Lessonsのスライド単位で学習状況をリアルタイムに把握できる

2020年度の授業アンケート結果

- 難易度・分量・説明方法・教材・理解度を深める配慮・シラバス・学習成果・満足度の観点で全授業終了後にアンケート（「1. 全く当てはまらない」「2. あまり当てはまらない」「3. やや当てはまる」「4. よく当てはまる」の4件法）を実施
- モデル授業（後半はプログラミング演習）を実施した14クラスを対象



2020年度の自由記述による感想（一部抜粋）

- パワーポイントやEdなどの教材が体系的に作られており、理解しやすかった。
- 通常時でも半分はメディア授業を実施するという授業の特性からか、非常にスムーズに講義を受けることができました。講義動画のみではなく、演習も織り交せてくださっていたので、理解がしやすくなった部分が多くありました。
- 動画やファイルなどを使った分かりやすい説明と、楽しいクイズや実習といった様々な方法で授業を進めていただけたので、モチベーションを保ちながら勉強することができました。
- 動画形式での授業について、一度で理解できなかった部分が気軽に何度でも見直すことができ、理解がしやすいように感じた。そのため、今後もこの形式がメインでなくともこの形式を取り入れた授業形態にして欲しいと感じた。

**教材や授業設計の工夫次第でオンライン授業でも高い教育効果が
得られることが示唆された**



まとめ

- 2020年度春学期の初年次必修の全学共通教育科目の一般情報教育科目「情報社会基礎・情報科学基礎」の実施内容を報告
- 授業コンテンツの調整および同期・非同期の活動・支援をうまく組み合わせることにより、オンライン授業環境下でも授業改善を図ることができた

- **動画教材の活用**

→ 受講生安心して授業に取り組むことができる。復習にも効果的。

- **オンライン環境における演習の効果**

→ 受講生のモチベーションの維持、体験的な学習の効果。

- **受講生の学習状況をリアルタイムで把握できる仕組み（LA）**

→ Edのようなで受講生の学習状況を把握できる仕組みやLMSのクイズ機能等を活用

本授業で利用した教材の公開について



本授業の教材（講義動画、演習ツール）をOER（Open Educational Resources）として2021年1月に公開を予定しております。



<https://csedu.ime.cmc.osaka-u.ac.jp/oer/>