

AT Column & Fitting Note

高等教育アクセシビリティプラットフォーム（以下、HEAP）では、障害のある学生の権利を考えるための活動に取り組んでいます。障害学生支援分野のスタンダードを構築することを目的とし、高等教育における障害と支援に関する情報やノウハウを集約・発信しています。

Assistive Technology（AT：支援技術）の活用は、障害学生が学びの機会を得るために必要な代替手段を実現するために役に立つものです。しかし、これまでの障害学生支援の現場では、AT機器の使用は非常に限定的なものでした。HEAPではATの体験・貸出事業に加えてAT導入・活用について情報を発信しています。ぜひご活用ください。

ATに関する取り組み

- ATライブラリー
- Tips + 動画
- Fitting Note
- AT Column

Fitting Note

HEAPでは日々多くの相談が寄せられます。中には、AT（支援技術）の活用を提案することも少なくありません。そのような事例を一部改変して「Fitting Note」として、ディスアビリティに対するアセスメントや有効なATの選択肢を例示しながらまとめています。

1. 発達障害学生と板書
2. 上肢麻痺と筆記の代替
3. 全盲（重度視覚障害）とテキストデータ化



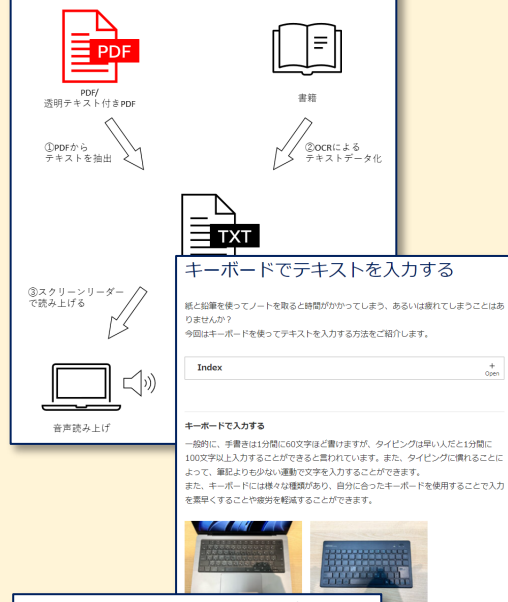
AT Column

「AT Column」では、AT（支援技術）の活用方法や導入の際に躓きやすい点などを障害学生支援の実務者目線で紹介しています。

1. 「Notability」を使ってiPadでノートを取る
2. iPadを使った音声認識とマイクを選ぶ際のポイント
3. 講義音声を録音する
4. マークアップを使った資料への書き込み
5. OneNoteで録音しながらノートを取る（タイピング）
6. スタイラスペンとタブレット端末を使った手書き入力
7. キーボードでテキストを入力する
8. スマートフォンで文書作成
9. 音声認識を使ってテキストを入力する
10. スイッチとソフトウェアキーボードを組み合わせたテキスト入力
11. マウス等の入力装置とソフトウェアキーボードを組み合わせたテキスト入力
12. 透明テキスト付きPDFからテキストデータを作成する
13. OCRソフトを使って書籍など印刷された資料のテキストデータを作成する
14. スクリーンリーダーを用いてPC等の端末でテキストデータを読み上げる
15. テキストデータを点字モバイル端末で読む
16. テキストデータを点字データに変換する
17. 点字プリンターを使って点字データを点字で印刷する
18. テキストデータから頻出単語を一括で抽出する
～音声認識の単語登録を簡単にする方法～



アクセシブルなデータへの加工フローの一例



スイッチとソフトウェアキーボードを組み合わせたテキスト入力

上記の順番によってキーボードやマウスの使用が困難な場合でも、スイッチとソフトウェアキーボードを組み合わせて使用することでテキスト入力が可能です。今回はWindowsやMac、iPad、Chromebookでスイッチを使ってテキストを入力する方法をご紹介します。

一般的に、手書きは1分間に60文字ほど書けますが、タイピングは早い人だと1分間に100文字以上入力することができると思われています。また、タイピングに慣れることにより、筆記より少ない種類の文字を入力することができます。また、キーボードには様々な種類があり、自分に合ったキーボードを使用することで入力を早くすることや疲労を軽減することができます。

スイッチとインターフェース

・スイッチ

スイッチには様々な形、大きさのものがあります。また、押下する時に少ない力で反応するようなスイッチもあります。指先や手のひらで押すスイッチの他に頬で押すスイッチもあります。



ジェリーピン

```

HWE (HEAP Word Extractor) .ipynb
ファイル 編集 実行 ランタイム ツール ヘルプ 最終編集: 6月27日
+ コード + テキスト
[] #単語を取得
word = node.surface
word = word.replace("、", "").replace("。", "")
#品詞を取得
pos = node.feature.split(",")[0]
#読みを取得
sound = node.feature.split(",")[7] if len(node.feature.split(",")) > 7 else ""

minimum_length_of_sound = 2
if (pos == "名詞" and len(sound) > minimum_length_of_sound):
    sound_hiragana = conversion.do(sound)

if word in word_list and word not in frequently_appearing_word_list:
    frequently_appearing_word_list.append(word)
    frequently_appearing_word_and_sound_list.append([word, sound_hiragana])
else:
    word_list.append(word)
  
```