

超速學習

EXAMPLES

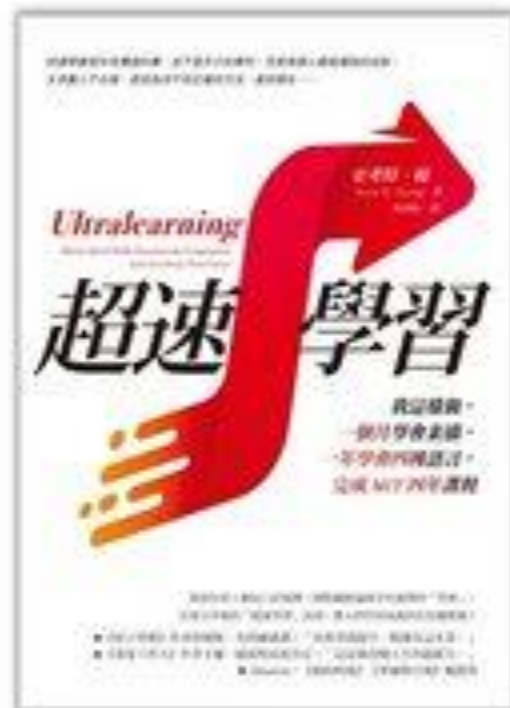
- 1. SCOTT H. YOUNG 本人1年學完4種語言/1年學完MIT資訊課程
 - 2. 超速學習的白老鼠-蒙特貝洛

教育制度下的 學習遷移

《超速學習》——一張圖掌握·學得快又好的9大法則

閱讀前哨站
READINGOUTPOST

《原子習慣》作者盛讚：真希望我能早一點擁有這本書！



9. 培養直覺

5. 保留記憶

6. 反覆操練

7. 專心致志

4. 提取記憶

2. 直截了當

8. 意見回饋

3. 勇於實驗

1. 後設學習

後設學習=學習地圖(百分之十)省下白工時間

1.Why What How>>功能性與本質性的區分目的

- 避免學習錯誤

- 2.專家訪談法>>看專家如何達成(標竿)以及請專家看你的學習計畫

- 3.列出需要理解的事 需要記憶的事 需要練習的事

- | | | |
|---|----|----|
| 牌 | 規則 | 題庫 |
|---|----|----|

- 4.發現效益遞減時，在後設學習一次

直截了當(走最短的路)

- 直接學習與你要用的技能有關的能力(與環境有關)

反覆操練

- 1. 奧運選手
- 2. 程式達人 William Lin

作者建議：先直接學習再反覆操練

在心理上感到費力的事，比感到輕鬆的事還能為學習帶來更多好處。

提取記憶-利用測驗來學習

- 錦堂：複習式筆記(費曼學習法)，錯誤式筆記
- 建議：有些事若無法精通，只要知道要怎麼學會它就好

意見回饋

- 1. 質量比數量重要
- 2. 挑戰要難，但不能太難

培養直覺

- 1.別輕易看詳解來放棄困難題目
 - 2.利用證明過程來理解
 - 3.利用具體例子來學習(比喻)你如何思考它
-
- 本章補充說明：
 - 1.看待天才-世人過多聚焦在結果，而非過程
 - 2.直覺是大量有條理處理問題經驗的產物

勇於實驗

- 1.先模仿再創造(藝術家為例)自身例子：汲取他人的優點於一身
- 2.把各種方法放在一起比較
- 3.採用新的限制(學到後面跳脫習慣做事)

補充

- 1.第一資訊飲食法
- 2.愉悅來自於擅長某一件事
- 3.自發性追求，非外在責任
- 4.完成學習的終點會看到更多可能性
- 到了夢想，就會有更大的夢想
- 5.我的夢想

Note: By applying to this position your application is automatically submitted to the following locations: **Taipei, Taiwan; Xindian District, New Taipei City, Taiwan**

- Bachelor's degree in Computer Science, Electrical Engineering, related technical field or equivalent practical experience.
- Software development experience in one or more general purpose programming languages such as: C/C++, Java or Python.
- Experience in the field of software engineering (i.e. working with data structures, algorithms, software design).

- PhD in Computer Science, Electrical Engineering, or related field.
- 5 years of relevant experience in one or more of the following: embedded systems, system bring-up, device drivers, multimedia, browser development, camera algorithm/image quality, ISP development, video encoding/decoding, audio/video playback, DRM technologies, tools development, browser technologies, applications, Networking and connectivity, and smartphone/computer security.
- Experience with concepts in computer architecture and Linux system programming.
- Knowledge of one or more of the following: Chromium code base and APIs, graphics.
- Ability to speak and write in English fluently and idiomatically.

This site uses cookies from Google to deliver its services and analyze traffic.

Learn more

OK

資工課程尋找

- 1.youtube
- 2.ocw
- 3.MIT
- 4.MOOCs

尋找高手

- 1.詢問學姊>>推薦MIT課程
- 2.不會問同學
- 3.先做再說(先學C++)後來發現要先學C
- 4.打電話問在業界工作的親戚(推薦買書看)

- C看玩C++ JAVA PYTHON 面試題庫
- 演算法