

デザイン・ コンピューティング入門

—Python による建築の形態と
機能の生成・分析・最適化—

日本建築学会 編

コロナ社

本書関係委員

(2017年3月) 五十音順・敬称略

日本建築学会 情報システム技術委員会

委員長 倉田 成人
幹事 池田 靖史 猪里 孝司 大崎 純 下川 雄一
委員 (省略)

デザイン科学教育方法研究小委員会

主査 藤井 晴行
幹事 大崎 純 長坂 一郎
委員 位寄 和久 岩田伸一郎 川角 典弘 小林 祐貴
藤田慎之輔 水谷 晃啓 安田 溪 渡辺 俊

執筆担当者

第1章 渡辺 俊 藤井 晴行
第2章 2.1 藤田慎之輔 大崎 純 安田 溪
2.2 藤田慎之輔 大崎 純
2.3 大崎 純 藤井 晴行
2.4 大崎 純
2.5 小林 祐貴
第3章 3.1 渡辺 俊
3.2 藤田慎之輔 大崎 純
3.3 小林 祐貴
第4章 4.1 渡辺 俊
4.2 小林 祐貴
4.3 安田 溪
第5章 藤田慎之輔 大崎 純
第6章 長坂 一郎
第7章 7.1 藤田慎之輔
7.2 水谷 晃啓
7.3 水谷 晃啓

はじめに

本書では、プログラム言語 Python を用いて建築の研究や設計に用いられる代表的な数理的手法のいくつかを解説します。私たちは数理的手法を使いこなして建築の研究や設計に活用したいと思っている人たちを対象に、本書を執筆しました。形態生成、最適化、知識処理などの数理的手法の要点を説明し、それらを Python によってプログラミングする方法を解説します。数理的手法の説明を読み、自分の手でプログラムを書いてみるのが数理的手法を使いこなすための一歩になると考えています。

今日、さまざまな数理アプリケーションを手軽に用いることが可能です。内部でどのような計算をしているのかを知らなくても、データを入力すると答えらしきものが出力されます。どのような計算がなされているかを知ると、アプリケーションを適切に使用したり自由自在に使いこなしたりすることが可能になります。自動車の運転に例えてみましょう。自動車の仕組みや構造を詳しく知らなくても、運転することは可能です。しかし、自動車を賢く使いこなすためにはこれらをよく知っていることが肝要です。アプリケーションの利用についても同様のことが言えます。

本書を用いた学習のために高価なアプリケーションを入手する必要はありません。Python は数理的計算に広く用いることができるプログラム言語です。Python を用いたプログラム作成の環境は無償で入手することができます。また、さまざまなライブラリが提供されていて、これらを用いることによって、より高度で本格的なプログラムを作成することも可能です。なお、本書に登場するプログラムは Web ページからダウンロードすることができます (p.7 参照)。

本書は入門書です。学習を進めプログラムを開発していくと、本書では物足りないと感じるようになると思います。それは私たちにとって嬉しいことです。読者の皆さんが本書を入り口として、本格的な数理的手法を用いた研究や設計の世界に入ることを望んでいます。

本書が皆さんのお役に立てれば幸いです。

2017年7月

日本建築学会 デザイン科学教育方法研究小委員会

本書における、Microsoft, Windows, Python, Blender, Grasshopper ほか、記載されている会社名、商品名、製品名は、一般に各社の登録商標、商標、または商品名です。本文中では、TM, ©, ® マークは省略しています。

目 次

1 デザイン科学とコンピューティング

1.1	デザイン科学とは	1
1.2	デザインにおける思考とは	2
1.3	プログラミングを学ぶ意味	5

2 Python 入門

2.1	Python の概要と基本操作	8
2.1.1	Python の概要	8
2.1.2	Python 本体と各種ライブラリのインストール	8
2.1.3	基本的な操作と演算	10
2.1.4	条件分岐と繰り返し処理	13
2.1.5	関数の定義	15
2.1.6	スクリプトファイルの作成とファイルの入出力	16
2.2	ライブラリの利用	17
2.2.1	標準ライブラリの利用	17
2.2.2	matplotlib を用いたグラフの作成	21
2.2.3	NumPy/SciPy を用いた数値解析	23
2.2.4	ユーザーライブラリの作成	25
2.3	再帰プログラミング	26
2.4	クラス	29
2.5	CAD・CG ソフトウェアとの関係	30
2.5.1	Python で絵を描く方法	30
2.5.2	Blender のインストールと Python スクリプトの実行	31
2.5.3	図形の配列	34
2.5.4	フラクタル図形の生成	35

3 形態の生成

3.1	形態文法	40
-----	------	----

3.1.1 定義	40	3.1.2 CGI の実装	42
3.1.3 建築デザインへの応用	44	3.1.4 日仏タイルの形態文法	45
3.2 形態の表現手法	50		
3.2.1 ベジエ曲線	50	3.2.2 非一様有理Bスプライン	54
3.2.3 パラメトリック曲面	54		
3.3 計算幾何学	57		
3.3.1 計算幾何学とは	57	3.3.2 基本的なアルゴリズム設計技法	57
3.3.3 符号付き面積	58	3.3.4 凸包の描画	58
3.3.5 ボロノイ図の定義と性質	61	3.3.6 逐次添加法によるボロノイ図構成アルゴリズム	61
3.3.7 Python を用いたボロノイ図の描画	62		

4 分 析

4.1 グラフ・ネットワーク	64		
4.1.1 定義	64	4.1.2 最短経路問題	65
4.1.3 スペースシンタクス	70	4.1.4 最近傍探索	77
4.2 列 挙	80		
4.2.1 列挙問題とは	80	4.2.2 多面体の展開のグラフ問題への対応付け	80
4.2.3 アクセスグラフの全域木	84		
4.3 待ち行列シミュレーション	86		
4.3.1 建築における行動シミュレーションと待ち行列	86	4.3.2 α 法と β 法—建築計画における規模算定に用いられる待ち行列理論	87
4.3.3 進め方—シンプルなモデルからはじめよう	88	4.3.4 人が行動するプログラムの骨格	88
4.3.5 1人が1つのトイレを利用する	90	4.3.6 複数人が1つのトイレを利用する	94
4.3.7 SimPy を利用する	98	4.3.8 記録とその可視化をしながら、複数人で複数個のトイレを利用	103

5 形態と性能の最適化

5.1 最適化問題とは	107		
5.1.1 最適化問題の定義	107	5.1.2 最適化手法	109

5.2 建築の最適化	117	5.2.1 建築の最適化問題の概要	117	5.2.2 建築の最適化問題の例	117
5.3 形態の最適化	124	5.3.1 最適化手法を用いた建築形態設計の概要	124	5.3.2 形態最適化の例	125

6 デザインに関する知識の処理

6.1 知識表現	139	6.1.1 デザインと知識表現	139	6.1.2 知識とは？	139
		6.1.3 表現とは？	140	6.1.4 推論とは？	140
6.2 論理に基づいた知識表現	141	6.2.1 命題論理	141	6.2.2 トートロジー	143
		6.2.3 推論	144	6.2.4 述語論理	145
		6.2.5 述語論理による知識表現	146	6.2.6 推論	147
		6.2.7 プログラム	147		
6.3 意味ネットワーク	149	6.3.1 継承による推論	151	6.3.2 多重継承の問題	151
		6.3.3 プログラム	152		

7 コンピュータショナル・デザインの事例

7.1 Python を用いたテンセグリティの形状決定と施工事例	157
7.2 「東京計画 1960」のオフィス群配置デザインの再現プログラム	165
7.2.1 「東京計画 1960」の概要	165
7.2.2 丹下モジュールと「東京計画 1960」	166
7.2.3 オフィスエリアの概要とそのシステム	167
7.2.4 Python スクリプトの実行結果とその解説	170
7.3 その他の事例	176
7.3.1 形状決定ツールとしての応用例	176
7.3.2 デジタルファブリケーションツールとしての応用例	177

引用・参考文献	179
索引	181

1

デザイン科学と コンピューティング

本章では、デザインとは何か、デザイン思考とは何かについて考えるとともに、デザインにおけるコンピュータ利用について考える。特に、デザインと科学の関係を論理的に解説するとともに、デザインの修養においてプログラミングを学ぶ意味について解説する。

1.1 デザイン科学とは

「デザイン」とは、人によってその理解するところが異なる言葉であろう。多くの人はまず、ポスターやパッケージなど、表層的な図案や意匠のことを思い浮かべるかもしれない。あるいは、自動車や家電品など、製品全般にまで視野を広げ、機能的な意味も含めて「形態」を構想することと理解するかもしれない。

工学分野の人であれば、設計と同義語と考えるだろう。一方、政策デザイン、キャリアデザイン、ライフデザインのように、物理的な形態を伴わない場合にもデザインという言葉が用いられる。

さらに最近では、社会デザイン、情報デザイン、行動デザイン、コミュニティデザインといった言葉も使われており、デザインの外延はますます拡大し、さらにその意味するところが何なのかわかりづらくなっているように思われる。

ノーベル経済学賞を受賞したハーバート・サイモン (H. Simon) は、著書「システムの科学¹⁾」の中で、「現存の状態をより好ましいものに変えるべく行為の道筋を考案すること」がデザインであるとした。有形・無形に限らず、われわれの社会を取り巻くすべての人工物 (artifacts) は、目的に供するように試行錯誤した結果生み出されたものであり、その意思決定こそがデザインだとしたのである。

すなわち、デザインとは、一部で誤解されるような、センスのある人が上手に絵を描くことだけや単なる思い付きのみに立脚した発想の類ではなく、そこには望まれる結果を達成するための深い思慮が伴わなければならない。

「科学」もまた、人によってその理解するところが異なる言葉である。多くの人はまず、廊下の突き当たりの特別教室での実験授業の思い出から、理科教科をイメージするかもしれ

† 肩付き数字は、巻末の引用・参考文献の番号を表す。

ない。広義には、自然科学、人文科学、社会科学という言葉からも連想されるように、それは学問の基盤をなすものであり、それぞれの学問領域に合った独自の方法をもっている。一般的には、自然科学のように、主観的ではなく客観的な思考に基づく知識の体系と考えられるだろう。

一方で、何をもって客観的と認めるかは、背負っている立場により見解の相違がありそうである。例えば、人工知能研究の領域には知能や知性を一人称の視点から捉えようという動きもある²⁾。

ノーベル物理学賞を受賞したスティーヴン・ワインバーグ (S. Weinberg) は著書「科学の発見³⁾」の中で、科学とは「観察」「実験」「実証」に基づかなければならず、自然の観察抜きに論理や数学だけに頼って理解しようとするのは科学ではないと断じた。その意味では、アリストテレスやデカルトなど数多の科学史の偉人達も詩人にすぎず、科学と呼べるのは物理学を筆頭に化学や生物学くらいまでで、それ以外は数学も含めて科学ではないと切り捨てている。主張に賛否はあるものの、何にしても科学を標榜する以上は、対象を広く深く洞察し、それらを体系的に記述・解釈・説明できる理論 (モデルや規範) を構築・提供することが求められるといえるであろう。

このように考えると、「デザイン科学」とはきわめて難儀な言葉であることがわかる。

デザインという言葉の名詞として捉えるならば、デザイン科学とは、先人がデザインした数多の人工物の丹念な観察から、その背後に隠された論理を導き出すことと翻訳できる。美しいプロポーションに共通するとされる黄金比への言及などは、その真偽は別として、最も古典的なデザイン科学といえるかもしれない。クリストファー・アレグザンダー (C. Alexander) の「パターン・ランゲージ⁴⁾」も「無名の質」を備えた空間に共通する特質を論理的に分析する中から、パターンを導いている。

一方で、デザインという言葉をも動詞として捉えるならば、デザイン科学とは、人間のデザイン行為あるいはデザイン思考そのものの論理の構築と解釈できる。無限の組み合わせから目的に見合うさまざまな可能性を網羅的に検討し、その中から最適な解を導き出すための方法論といえるかもしれない。建築家の菊竹清訓は、自らの建築設計プロセスを深考し、いかに合目的にデザインを進めるべきかに関する方法論を「か・かた・かたち⁵⁾」として総括した。工学者の吉川弘之は、デザインの可能性と限界を論理的な推論に基づく「一般設計学」として体系化している⁶⁾。

1.2 デザインにおける思考とは

では、論理的思考とは、いかなる思考なのだろうか。論理学の世界では、推論には、えんえき 演繹推論 (ディダクション)、帰納推論 (インダクション)、発想推論 (アブダクション) という

3つの形式があるとされている。これらは、下記のように定式化できる⁷⁾。

演繹推論： 事例 + 規則（ルール）→ 結果

帰納推論： 事例 + 結果 → 規則（ルール）

発想推論： 規則（ルール）+ 結果（目標）→ 事例

演繹推論とは、前提条件（事例）が与えられたときに、既知の規則（ルール）を適用して、結果を導く推論である。導かれるものごと（演繹推論の場合は結果）の論理的な正しさを保証する唯一の推論形式である。古くは、アリストテレスにより三段論法として定式化されたもので、「ソクラテスは人間である」という事例（小前提）と、「人間は必ず死ぬ」という規則（大前提）から、「ソクラテスは必ず死ぬ」という結果を導く例が示すように、演繹推論は唯一の論理的に正しい結果を導き出す。算数の授業で足し算の方法（ルール）を習った子供が、例えば $26 + 38 = \square$ のような計算ドリルをしている状況を想像してみよう。計算ドリルの答えは、足し算の方法を正しく用いれば 64 であり、それ以外は不正解である。

帰納推論とは、事例と結果が観察されたときに、事例と結果を結びつける規則（ルール）を導き出す推論である。子供は、事前に厳密な文法（ルール）を習うわけでもなく、周囲の会話を聞き取る中から母国の言葉を習得していく。ニュートンが、リンゴが木から落下するのを見て万有引力を発見したとされる俗説は、誰でも一度は聞いたことがあるであろう。

賢者の帰納推論結果の蓄積こそが科学的知識であり、われわれはその恩恵として演繹推論のための規則（ルール）を勞せずして知り、正しい答えを導くことができる。ただし、帰納推論は導かれる規則（ルール）の論理的な正しさを保証する推論形式ではない。導かれる規則（ルール）の正しさは観察されている事例と結果をふまえた蓋然的^{がいぜん}なものである。新たに観察される事例と結果が既存の規則（ルール）では説明できないとき、規則（ルール）の変換を迫られる。例えば、相対性理論はニュートン力学では説明できない事例と結果を説明すべく考案された。

発想推論とは、望まれる結果（目標）が与えられたときに、既知の規則（ルール）から遡って事例を導き出す思考である。先の算数の授業との対比として、 $\square + \square = 64$ のような計算ドリルに取り組む欧米の授業を紹介したテレビコマーシャルが話題になったことがある。もちろん不正解は多々あるが、発想推論は演繹推論のように正解が1つとは限らない。創造力を伸ばすには、このような解の候補をあげてはその正否を確かめつつ解を見つけるといふ試行錯誤（トライ・アンド・エラー）の積重ねが大切であるとされ、その過程で負の数の存在に気づくかもしれない。

建築設計競技（コンペ）において、同じ建築設計条件（目的）にもかかわらずさまざまな計画案が出てくるのも、それぞれの建築家が試行錯誤した結果であろう。このように考えると、デザイン思考とは、まさに発想推論（アブダクション）であることがわかる⁸⁾。

吉阪隆正（U研究室）は、御茶ノ水の神田川沿いに建設された旧日仏会館が雑誌「近代建築（1960年6月号）」で紹介されるにあたり、「コンビネーションとパーミュテーション」と題する以下のような一文を寄せている⁹⁾。

鍵をさずけよう。秘密を解く鍵を。

日仏両国の結びつきの妙なることを形にあらわして見たかつた。その鍵を。

〈中略〉

このタイルをよく眺めて貰いたい。そこに鍵はかくされているのだ。それは日仏をローマ字でFとJとして組合せてある。色がわりはあるにしても 何れも同じFとJである。但し FとJとは必ずしもFJとなっていないで FJとも JFとも そして又fjともなっているのだ。

ここで数学の順列組合せを思い出して欲しい。たつた一つのFJのタイル。これをいろいろに組合せたら面白い図柄になるだろうと考えたのが出発だったが 意外にもその組合せの多いのに驚き 到底皆やつて見てその中から選ぶということが不可能なのをさとつたのだつた。

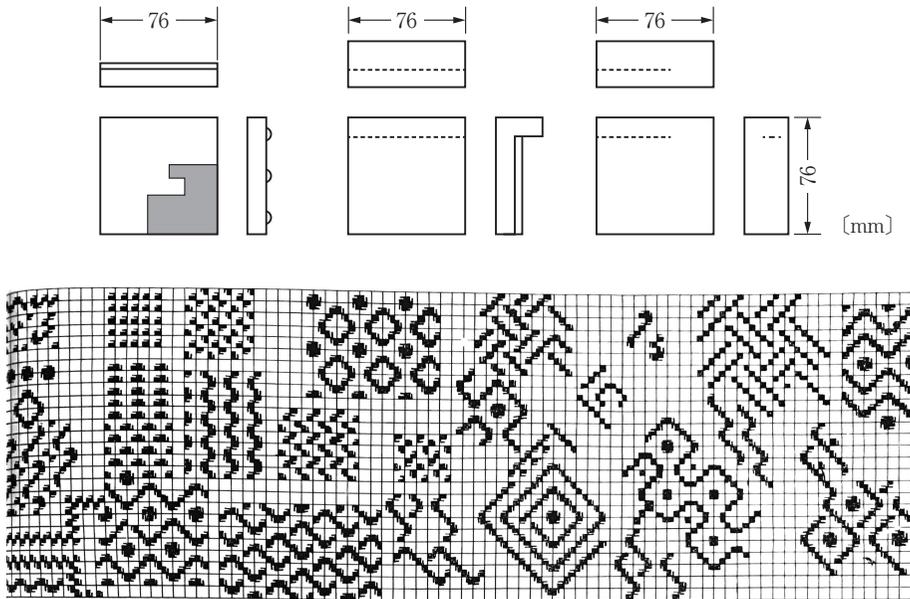
〈中略〉

だが考えて見ると建築の設計それ自身が この順列組合せのように思えてくる。そして建築の設計の場合には このタイルのように規格ではなくて 一つ一つがそのタイルの中でさえ変化するような それでいて何かの函数になっているような そんなものの順列組合せなのだ。それは人生からくらべると全く無限の宝庫だ。別な意味でいうと自由すぎる。私たちはその中で 時間の許す限り 力の許す限りのものを抜き打ちにやつて見て選ぶ外ない。常識は過去に経験ずみの順列組合せを後生大事に抱いて廻る。それは無数にあるものの中のたつた一つであるのに。そしてもつともつと具合のよい 美しいものもあるのにそれを顧みようとしない心である。

だが 他の順列組合せが 全部美しいともいえないし 具合がよいともいえない。大部分は無統一な 乱脈さを露呈する。だから一度見つけた美しい秩序を失いたくないというのはよくわかる。何億とある中の一つなのだから。それを敢えて犯し 新たなるものを発見するには勇気を要する。その勇気を暗示する意味でも 私達はこのFJのタイルの鍵を用いたい。それは単に日仏の交流ということに止まらず もつともつと広い世界を私たちは前途に持っていることを示してくれるからだ。

今私達は鍵を教えた。一つ使つて新世界を 未知の世界を発見し 打ちたてて見てはどうでしょう。私たちも大いに協力致しましょう。

記事に添えられた試行錯誤時のスケッチには、手書きのマス目に手当たり次第のパターンが描かれており、U研究室の大竹十一がその推敲に数か月を費やしたという苦勞の痕跡を伺うことができる（ 1.1)。では、鍵を受け渡されたわれわれは、どこまでデザインの可能性を探求し、未知の世界を発見できているだろうか。（詳細は「3.1.3項 建築デザインへの応用」を参照）。

図 1.1 FJ タイルと配置パターン⁹⁾

1.3 プログラミングを学ぶ意味

かつて大竹十一を悩ませた手作業に代り、いまはコンピュータによる生成がある。デザインに関心のある本書の読者の多くは、物心がついたときにはすでにコンピュータが身の回りにあたりまえのように存在しており、コンピュータを使うこと自体には抵抗のないデジタルネイティブと呼ばれる世代であろう。その意味では、コンピュータの基本的な使い方に戸惑ったりすることはないかもしれない。では、自身の情報リテラシーは、どのレベルだろうか？

いまや、ワープロや表計算のような一般的なアプリケーションは、中高生でもそれなりに使いこなしている。一方で、専門的なアプリケーションを使うには、単にアプリケーションの使い方だけではなく、その分野の専門知識が求められる。統計ソフトを正しく利用するには統計学の知識が必要になるし、CAD を使いこなすには設計製図の知識が必要である。これらの専門知識は、一般に大学などで学ぶことになるだろう。

しかし、せっかくのコンピュータを使いながら、コピー・アンド・ペーストで適当にすませてはいないだろうか。あるいは、インタラクティブな操作に長けただけで満足してはいないだろうか。確かに、既成の CAD/CG ソフトを利用して綺麗な図面やドローイングは簡単に作成できるかもしれない。しかし、それでは手作業の時代と大差はない。いや、これでは無限の宝庫の探求どころか大竹十一の手作業に全く及んでいない。

索 引

【あ】	
アイソピスト	70
アクセスグラフ	84
【い】	
遺伝的アルゴリズム	115
意味ネットワーク	140, 149
【え】	
エージェントモデル	94
【お】	
オブジェクト	29
重み付き線形和法	108
【か】	
関数記号	146
完全グラフ	65
【き】	
疑似焼きなまし法	116
行 列	87
極小曲面	133
許容解	107
許容領域	107
距 離	70
【く】	
クラス	29
グラフ	149
グラッドストラクチャ法	124
【け】	
計算幾何学	57
継 承	151
形状最適化	121
形態文法	40
経 路	64, 70
【こ】	
格子グラフ	84
恒真式	143
構造最適化問題	120
勾配ベクトル	109

公 理	144
コンプライアンス	125
【さ】	
再帰プログラミング	26
最急降下法	109
再帰呼出し	26
最適性条件法	114
サブクラス	29
【し】	
ジェネレーター	98
自己相似	35
自己釣合い軸力	158
指数分布	91
施設配置最適化問題	117
自然演繹	144
シミュレーション	88
述語記号	146
述語論理	140
ジョイント・コア・システム	167
証 明	144
真理値	142
真理値表	142
【す】	
推 論	140
推論規則	140, 144
数理計画法	109
スター	70
スプライン曲線	50
スプライン曲面	50
スペースシンタックス	70
【せ】	
制御多角形	50
制御点	50
制御ネット	50
制約法	108
宣言の知識	139
全称記号	146
全称例化	147
全体剛性行列	128

【そ】	
存在記号	146
【た】	
ダイクストラ法	65
タブー探索法	116
タプル	12
多目的計画法	108
多目的最適化問題	107
単一目的最適化問題	107
丹下モジュール	166
【ち】	
逐次構成法	57
逐次2次計画法	115
知識表現	139
知識表現と推論	139
【つ】	
ツリー	65
【て】	
定義多角形	50
定数記号	146
定 理	144
手続の知識	139
テンセグリティ	157
テンソル積ベジエ曲面	54
【と】	
東京計画 1960	166
凸 包	58
トポロジー最適化	121
【に】	
2部グラフ	65
【は】	
発見の手法	109
パレート最適解	108
バーンスタイン基底関数	50
【ひ】	
非一様有理スプライン	54

菱形継承問題	152	待ち行列理論	87		
ビジュアルプログラミング	170			【れ】	
		【め】		列挙問題	80
【ふ】		命題	141		
符号付き面積	58	命題記号	141	【ろ】	
部材剛性行列	128	命題論理	141	論理記号	141, 146
フラクタル図形	35			論理定数	141, 146
		【も】			
【へ】		モーダスポーネンス	140	【欧字】	
閉路	64			BIM	139
ベジエ曲線	50	【ゆ】		CAD	30
ベジエ曲面	50	有向グラフ	149	CG	30
ハッセ行列	115	ユーザーライブラリ	25	GA	115
ペナルティー法	112			has-a 関係	150
変数記号	146	【ら】		Int.V 値	70
		ラインサーチ	110	is-a 関係	150
【ほ】		ラグランジュ関数	114	kd-tree	78
ポアソン分布	90	ラグランジュ乗数	114	made-of 関係	150
母点	61	ラグランジュの未定乗数法	114	NURBS	54
ボロノイ図	61			part-of 関係	150
ボロノイ辺	61	【り】		RA 値	70
ボロノイ領域	61	リスト	12	RRA	70
		量化記号	146	SA	116
【ま】		量子化	146	TS	116
待ち行列シミュレーション	86			ZDD	85

【Python 関連用語】

Anaconda	8	jit	136	random	18
csv	19	lambda	24	return	15
def	15	math	17	Rhinoceros	170
for	14	matplotlib	21	SciPy	24
generator	99	numba	136	SimPy	98
Grasshopper	170	NumPy	23	turtle	19
if	13	open	16	while	14
import	17	pip	9		

— 著者略歴 —

藤井 晴行 (ふじい はるゆき)

1983年 早稲田大学理工学部建築学科卒業
1985年 早稲田大学大学院理工学研究科博士前期課程修了(建設工学専攻)
1985年 清水建設株式会社勤務
1994年 米国カーネギーメロン大学大学院人文社会科学研究科修士課程修了(哲学専攻計算言語学コース)
1998年 博士(工学)(早稲田大学)
1999年 東京工業大学講師
2000年 東京工業大学大学院助教授
2007年 東京工業大学大学院准教授
2015年 東京工業大学大学院教授
現在に至る

渡辺 俊 (わたなべ しゆん)

1985年 早稲田大学理工学部建築学科卒業
1987年 早稲田大学大学院理工学研究科博士前期課程修了(建設工学専攻)
1989年 早稲田大学助手
1993年 博士(工学)(早稲田大学)
1993年 早稲田大学講師
1994年 筑波大学講師
1998年 筑波大学助教授
2007年 筑波大学大学院准教授
2013年 筑波大学大学院教授
現在に至る

水谷 晃啓 (みずたに あきひろ)

2007年 芝浦工業大学工学部建築工学科卒業
2009年 芝浦工業大学大学院修士課程修了(建設工学専攻)
2009年 隈研吾建築都市設計事務所勤務(プロジェクト契約)
2010年 SAITO ASSOCIATES 勤務
2013年 芝浦工業大学大学院博士後期課程修了(地域環境システム専攻), 博士(工学)
2013年 芝浦工業大学博士研究員
2014年 豊橋技術科学大学大学院助教
2017年 豊橋技術科学大学大学院講師
現在に至る

藤田慎之輔 (ふじた しんのすけ)

2008年 名古屋大学工学部社会環境工学科卒業
2010年 京都大学大学院工学研究科修士課程修了(建築学専攻)
2010年 有限会社金箱構造設計事務所勤務
2012年 株式会社 Dawn of the New Architecture 共同設立(2016年 株式会社 DN-Archi に改組)
2013年 広島大学大学院工学研究科修士課程修了(建築学専攻), 博士(工学)
2016年 東京工業大学大学院助教
現在に至る

大崎 純 (おおさき まこと)

1983年 京都大学工学部建築学科卒業
1985年 京都大学大学院工学研究科修士課程修了(建築学専攻)
1985年 京都大学助手
1993年 博士(工学)(京都大学)
1996年 京都大学大学院助教授
2007年 京都大学大学院准教授
2010年 広島大学大学院教授
2015年 京都大学大学院教授
現在に至る

長坂 一郎 (ながさか いちろう)

1990年 早稲田大学理工学部建築学科卒業
1992年 英国建築協会付属建築学校修了
1993年 株式会社フェイズ計画研究所勤務
1998年 東京大学大学院工学系研究科博士課程修了(精密機械工学専攻), 博士(工学)
1999年 神戸大学助手
2003年 神戸大学大学院助教授
2007年 神戸大学大学院准教授
2015年 神戸大学大学院教授
現在に至る

小林 祐貴 (こばやし ゆうき)

2011年 京都大学工学部建築学科卒業
2013年 京都大学大学院工学研究科修士課程修了(建築学専攻)
2015年 東京工業大学大学院助教
現在に至る

安田 溪 (やすだ けい)

2012年 京都大学工学部建築学科卒業
2014年 京都大学工学研究科修士課程修了(建築学専攻)
2014年 京都大学工学研究科建築学専攻博士課程
現在に至る

デザイン・コンピューティング入門
—Pythonによる建築の形態と機能の生成・分析・最適化—

Introduction to Design Computing

—Python Programming for Generation, Simulation and Optimization of
Architectural Form and Function—

© 一般社団法人 日本建築学会 2017

2017年9月28日 初版第1刷発行

検印省略

編者 一般社団法人 日本建築学会
発行者 株式会社 コロナ社
代表者 牛来真也
印刷所 萩原印刷株式会社
製本所 有限会社 愛千製本所

112-0011 東京都文京区千石 4-46-10

発行所 株式会社 コロナ社

CORONA PUBLISHING CO., LTD.

Tokyo Japan

振替 00140-8-14844・電話 (03)3941-3131(代)

ホームページ <http://www.coronasha.co.jp>

ISBN 978-4-339-05254-1 C3052 Printed in Japan

(高橋)



JCOPY <出版者著作権管理機構 委託出版物>

本書の無断複製は著作権法上での例外を除き禁じられています。複製される場合は、そのつど事前に、出版者著作権管理機構（電話 03-3513-6969, FAX 03-3513-6979, e-mail: info@jcopy.or.jp）の許諾を得てください。

本書のコピー、スキャン、デジタル化等の無断複製・転載は著作権法上での例外を除き禁じられています。購入者以外の第三者による本書の電子データ化及び電子書籍化は、いかなる場合も認めていません。落丁・乱丁はお取替えいたします。